Simula Research Laboratory

– by thinking constantly about it

Aslak Tveito
Are Magnus Bruaset
Olav Lysne
Editors

Springer

Simula Research Laboratory

Aslak Tveito • Are Magnus Bruaset • Olav Lysne
Editors

Simula Research Laboratory

— by thinking constantly about it

Springer

Editors
Aslak Tveito
Are Magnus Bruaset
Olav Lysne
Simula Research Laboratory
PB 134 Lysaker
Norway
aslak@simula.no
arem@simula.no
olavly@simula.no
www.simula.no

ISBN 978-3-642-01155-9 e-ISBN 978-3-642-01156-6
DOI 10.1007/978-3-642-01156-6
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009938722

Mathematics Subject Classification (2000): 94C99, 65N21, 65M60, 68N99

The front cover: The employees are the foundation of Simula Research Laboratory. To symbolize this we created the front cover image from photos of employees. At Simula, a sculpture depicting the head of a thinking human is awarded to the "Researcher of the year." An image of this sculpture forms the basis of the montage.
Cover design: WMXDesign GmbH, Heidelberg

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

When researchers gather around lunch tables, at conferences, or in bars, there are some topics that are more or less compulsory. The discussions are about the hopeless management of the university or the lab where they are working, the lack of funding for important research, politicians' inability to grasp the potential of a particularly promising field, and the endless series of committees that seem to produce very little progress. It is common to meet excellent researchers claiming that they have almost no time to do research because writing applications, lecturing, and attending to committee work seem to take most of their time. Very few ever come into a position to do something about it.

With Simula we have this chance. We were handed a considerable annual grant and more or less left to ourselves to do whatever we thought would produce the best possible results. We wanted to create a place where researchers could have the time and conditions necessary to reflect over difficult problems, uninterrupted by mundane difficulties; where doctoral students could be properly supervised and learn the craft of research in a well-organized and professional manner; and where entrepreneurs could find professional support in developing their research-based applications and innovations.

This book is about what we did. It describes the philosophy behind Simula Research Laboratory, how we have tried to implement it, and the results that have come out of it. We have tried to describe this through a mix of articles and interviews. Most of the book is intended for anyone interested in research, whereas some of the articles are written for experts interested in Simula's research subjects.

At the time of this writing, Simula has existed for slightly more than eight years and it is fair to say that we still have a long way to go. The vision is still a vision, but by reading this book we hope you will sense a strong will to work towards this goal. We do not claim that we have found the solution but we are looking for it and will keep working towards that goal because our conviction is stronger than ever. We need researchers who are able to fully concentrate on their problems, we need to improve the way we educate new researchers, and we must become much better at developing applications based on our research.

Writing a book such as this is a big project involving many contributors. We would like to thank all of them for a fruitful collaboration. In particular, we would like to thank Arvid Hallén, Ingolf Søreide, Arild Underdal, Paul Chaffey, Odd Reinsfelt, Ine Marie Eriksen Søreide, Bjørn Rasmussen, Hans Christian Haugli, Morten Dæhlen, Hans Gallis, and Viktor S. Wold Eide for taking the time to be interviewed for the book. We would like to thank Christian Hambro for his introduction to Norwegian research politics and Bjarne Røsjø for his introduction to the difficult birth of IT Fornebu. Furthermore, we would like to thank Bjarne Røsjø and Dana Mackenzie for conducting all the interviews, Sverre Christian Jarild and Morten Brakestad for taking the photographs, and Anna Godson for translating from Norwegian. AcademicWord has been very efficient at editing the language of most of the chapters of the book and Marianne M. Sundet, Tomas Syrstad Ruud, and Åsmund Ødegård have done a superb job handling all the practicalities of the project. The front cover was designed by Åsmund Ødegård, Marianne M. Sundet, Are Magnus Bruaset, and Manfred Bender (WMXDesign GmbH) at Springer-Verlag. Finally, we wish to thank all members of Simula's staff who have contributed to this book in so many ways.

At Simula's opening, Dr. Martin Peters of Springer-Verlag attended as a guest alongside half the Norwegian government. Since then, Dr. Peters has been a frequent guest at Simula and all those visits have paid off handsomely for both parties. A total of ten Springer books have been produced by Simula employees. As always, our collaboration with Dr. Peters and with Springer-Verlag has been outstanding.

Special acknowledgements are due to the Norwegian Government represented by the Ministry of Education and Research, the Ministry of Trade and Industry, and the Ministry of Transport and Communications. It was a bold decision of them to create Simula, and we are grateful for the continuous financial and political support they have provided. We would also like to thank the Research Council of Norway for skillfully administering our relations to the Government.

Furthermore, we would like to thank StatoilHydro for their long-term commitment in research undertaken at Simula. Together with them we have built a strong activity on computational geosciences, and the yield of this collaboration forms the basis of several chapters of this book. We are also grateful for collaboration with and financial contributions from the Municipality of Bærum, Telenor, Det Norske Veritas, and Sun Microsystems.

If you have views, questions, or comments about the contents of this book, please do not hesitate to contact the authors of the chapters or Aslak Tveito at aslak@simula.no.

We hope you will find something to think about.

Fornebu, Norway, June 2009

Aslak Tveito
Professor
Managing Director
Simula Research Laboratory

Are Magnus Bruaset
Professor
Assistant Director
Simula School of Research and Innovation

Olav Lysne
Professor
Director of Basic Research
Simula Research Laboratory

Contents

Part I The Scene

Part II Basic Research

Part III Research Education

Part IV Research Applications

Some common abbreviations

CBC	Center for Biomedical Computing
CoE	Centre of Excellence
DNV	Det Norske Veritas
FFI	Norwegian Defence Research Establishment
ICT	Information and Communication Technology
Ifi	The Department of Informatics, University of Oslo
ND	The Network and Distributed Systems Department
NR	Norwegian Computing Center
NTNU	The Norwegian University of Science and Technology
RCN	The Research Council of Norway
SC	The Scientific Computing Department
SE	The Software Engineering Department
SI	Simula Innovation AS
SSRI	Simula School of Research and Innovation AS
SRL	Simula Research Laboratory AS
UiO	University of Oslo

PART I

THE SCENE

1

WHAT WOULD YOU DO IF YOU SUDDENLY GOT AN ANNUAL GRANT OF ABOUT TEN MILLION DOLLARS?

Aslak Tveito and Morten Dæhlen

Funding is one of the very few problems that every researcher on the planet has in common. Most will have to confront this burden repeatedly during their careers, and a dream of a grand solution is common amongst researchers. We have lived that dream and this book tells the story of how we have spent the grant we got.

The genesis of the huge grant was—believe it or not—the need for a new airport in Norway. When the Norwegian Parliament decided to shut down the national airport located at Fornebu, an extensive political process ensued to decide what should be done with the structure. After many rounds in the Parliament, it was decided that an IT centre should be established, which was to include a research lab with considerable basic funding. You will find more about the political turbulence surrounding this process in chapter 8.

The research lab was named the Simula Research Laboratory, after the programming language developed in Norway by Kristen Nygaard and Ole-Johan Dahl. Professor Morten Dæhlen was appointed as the first managing director, a post subsequently assumed by Professor Aslak Tveito after a hectic initial period. Our discussions of how a really good research lab should be organized and operated began in the 1980s

Aslak Tveito
Simula Research Laboratory

Morten Dæhlen
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_1,

when we (Aslak Tveito and Morten Dæhlen) worked together studying numerical analysis. Later, we both worked at SINTEF, a large research lab primarily run on industry funding. SINTEF is an extraordinarily well-operated institution, but the funding model is difficult if basic research is your prime interest. Following our stint at SINTEF, we were both hired as professors at the University of Oslo. Later, Morten Dæhlen left the university to become director of the division of natural sciences in the Research Council of Norway. Throughout our careers, our debate about how to best organize and run a research lab continued. If we were to establish a new research lab, we agreed that a close collaboration between the university, private companies, and existing research labs would be necessary in order to obtain the necessary funding.

An opportunity to test our ideas about to organize and run a research lab appeared more or less out of the blue. The IT centre at Fornebu was purely a political concept, and so was the idea of furnishing it with a research lab. We had never foreseen such an opportunity; how many times have politicians ever volunteered to form a new research lab? It was indeed a bold move to establish Simula, and this was spelled out clearly by the lab's advisory board:

> The Norwegian government and the research leaders who established Simula Research Laboratory (SRL) should be congratulated on their courage and foresight in crafting SRL. It is unusual these days for governments to take such a long-term view, to concentrate resources and to trust independent leadership. In this case it has paid off handsomely–establishing more rapidly than the founders could possibly have hoped an exceptionally successful institution, which has all the prerequisites for a research centre of considerable international importance.
>
> The Simula Scientific Advisory Board; July 2004[1].

Since its inception, Simula consistently has enjoyed strong political support and has been visited by many political delegations. Because Simula is a newcomer in the Norwegian research system, such support has been extremely important.

Given the opportunity, we seized it and started to implement our ideas that really were a true mix of a research lab (e.g., SINTEF) and a university. The platform Simula was built on can be summarized as follows:

1. After talking to active researchers in Norway and around the world, we had formed the impression that surprisingly few full-time researchers exist. Some have hefty teaching duties, some have to spend a large part of their time supervising students or young researchers, some spend a lot of time trying to obtain funding, and all complain about endless meetings. We wanted to create possibilities for researchers to focus more or less completely on their research.
2. The management model used in universities at that time had serious deficiencies. At the institutional level, the process of making a decision was very complex, time demanding, and hard to comprehend. It consisted of both formal and highly infor-

[1] The Simula Scientific Advisory Board consists of internationally renowned scientists; their role is explained on page 66.

mal hierarchies. In contrast, the SINTEF model was much clearer, more transparent, and more efficient; SINTEF was operated more or less like a private company. Thus, Simula followed the SINTEF method of management.

3. The process of hiring scientists at the university is overly bureaucratic. It was created with the best possible intentions–and it has resulted in employment for a large number of brilliant scientists–but it is extremely time consuming, thus the risk of loosing excellent candidates in the process is significant. Simula implemented a very efficient way of hiring that is based on headhunting. In particular, this was important in the early days when Simula had to deliver strong results within a few years.
4. In general, university researchers (in Norway) argue that the direction of research must be completely free of outside constraints, and most attempts to change this situation result in strong protest. The idea of research driven by the curiosity of the individual scientist is a good one that historically has generated a wealth of important results. However, other models are possible. At Simula, we decided that research should be more strongly directed than is the case at universities. This policy has enabled us to focus on a limited number of projects. Furthermore, many researchers find it reassuring to know that there is a plan for their activities. On the other hand, within each project and from day to day, each researcher has great freedom.
5. We have always felt a strong obligation to be able to explain to our politicians and to taxpayers at large *why* we are doing *what* we are doing. Thus, we focus on research projects with foreseeable useful applications. This approach does not discourage high-risk projects; indeed, few of our projects are likely to have a commercial outcome. However, the motivation for each of our projects lies in real-life situations. We do not support purely curiosity-driven projects, which distinguishes us from the Norwegian universities.
6. We strongly believe that research labs should promote collaboration with industry at large. Such collaborations were extremely important at SINTEF, but there the projects often were of short duration and had strict specifications. At Simula, we wanted to invite companies or other institutions to collaborate on long-term projects to address really challenging problems. In one particular case this has turned out to be successful; read more in chapter 40.
7. No one knows how to create business based on research. However, as part of a commercial IT centre, Simula was more or less expected to create new businesses, and this has been an important task since its inception. Examples of businesses that have been created are given in chapter 42, and these results are based on continued efforts to identify, support, and develop commercial ideas at Simula.

These seven ideas that constitute Simula's research platform have been refined through numerous discussions with many employees at Simula, with members of our Board of Directors, and with members of our Scientific Advisory Board. Our attempts to implement these ideas are discussed further in chapter 3. Certainly, these ideas are clearer now than they were initially, but their implementation must still be regarded as a beta-version. Our initial ideas were naïve and not based on understanding of the laws and regulations surrounding the day-to-day operations of limited companies; meeting the realities involved a dazzling series of challenges.

The efforts undertaken at Simula have, to a certain degree, impacted other parts of the Norwegian research system. Morten Dæhlen currently is the chairman of the Department of Informatics at the University of Oslo, where he applies ideas and experiences attained at Simula. Furthermore, the Research Council of Norway is about to evaluate the organizational model used at Simula to assess the merits of applying this model in other national research initiatives.

2

BY THINKING CONSTANTLY ABOUT IT

An interview with Aslak Tveito by Dana Mackenzie

In 2000, Aslak Tveito, the current Managing Director of Simula, and his predecessor Morten Dæhlen were handed a once-in-a-lifetime opportunity to build a new academic institution from scratch. Although Simula owes its existence to a political decision; it owes its scientific vision to Tveito, who has led the laboratory since Dæhlen's departure in 2002. Most of all, Tveito wanted Simula to be a place where scientists could do research, freed to the greatest extent possible from extraneous duties such as teaching, committee meetings, or the constant search for funding.

The commitment to research at Simula starts at the top. Before founding Simula, Tveito was the head of the research group in scientific computing at the University of Oslo—one of the three groups that came together under the aegis of the new laboratory. He has continued to be actively involved in the research of the Scientific Computing division of Simula, even while managing a large enterprise that now boasts 120 employees and a budget of more than 100 million Norwegian kroner. Though he is a manager by necessity, Tveito (a big fan of the *Dilbert* comic strip) still seems to be a Dilbert at heart.

In this interview, we talked with Tveito about his vision for Simula, how that vision has been put into practice. We also discussed how computing has come to be at the forefront of modern-day science.

"What is the story behind Simula's slogan 'By thinking constantly about it'? Where did it come from, and what does it mean?"

"The story is that many employees argued that we should have a slogan or that we should come up with the shortest possible description of what Simula is about. I thought that it was completely impossible. You would have to write a whole docu-

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_2,
© Springer-Verlag Berlin Heidelberg 2010

ment about it; and there would be no way to condense it into a slogan. In fact, I made several jokes about slogans. At that point I read a lot of *Dilbert*, and there were lots of slogan jokes in Dilbert, and I enjoyed them. So I didn't like the idea.

"But then I read an interview with Lennart Carleson by Björn Engquist, which was published by Springer in a book called *Mathematics Unlimited.* Carleson mentioned that he had seen a statement by Newton, where somebody had asked him how he came up with the law of gravitation. Newton said, "By thinking constantly about it." It occurred to me that this was really the essence of Simula: having a place where you can think constantly about something undisturbed, with no other duties."

"Why did you choose the name Simula? Are you concerned at all about people confusing the name of the laboratory with the computer language?"

"Not at all! There are not that many really well-known results from Norwegian scientists. But Norwegian scientists made a really significant contribution to object-oriented programming. Ole-Johan Dahl and Kristen Nygaard created this beautiful language, Simula, that was used at the University of Oslo for many years in the introduction to computer programming[1]. I learned programming in that language, and a whole generation did. It was a language that you could really find beautiful, very different from C and C++ which looked very ugly.

"It was my idea to name the organization after Simula and to call it Simula Research Laboratory. I remember exactly where I was sitting the first time I came up with the idea. I was at home, meeting with Morten Dæhlen, who was the first director (at that point the director-to-be) and I remember that I argued strongly that it should be Simula Research Laboratory. I had always wanted to work in a laboratory! In the beginning, of course, some people thought that we would like to start working on the Simula computing language again, but that was never the intention at all."

"Getting back to the business of full-time research... Now that you are an administrator, you can't do research full time, but what is the proportion of time that you spend on it?"

"I think I'm a 50–50 guy, and I think many people are in that position. Fifty per cent research and fifty per cent something else. Of course, teaching is the most common something else. I don't teach, but I supervise students, and I still have plenty of time to do research. I think that is related to the way Simula is organized. It is a well-functioning business, and we don't have disasters every other week or things that I have to clean up. It is a peaceful place and things are running smoothly, so there is no reason for me to be involved all the time. I don't really want to be. I'm not the kind of leader who wants to put my mark on every possible thing. I am perfectly happy seeing that someone is doing things totally different from what I would have done, and I wish them the best of luck!

[1] Simula is generally regarded as the first object-oriented programming language.

"I am told that Morten's office was in the library, right in the middle of things. But your office is at the end of the hall. Was that deliberate?"

"That changed one and a half years ago, after I went to San Diego for two months and just worked on my own. Up to then, I had been really in the middle of things, not in Morten's old office but an ordinary office up in the administration. It struck me that when I sat there and people saw me all the time, it was always so very easy for people to drop by. They knew if they had my decision on something, it would be the final decision, and there would be no more fuss about it. So it was very tempting for them just to drop by my office. But if you have six or seven or eight people just dropping by your office in a day, then your day is more or less destroyed. You cannot concentrate on anything for more than twenty minutes. I was really not satisfied with this, so I decided to move away from the front. I told everyone that before lunch I am doing mathematics, and after lunch we are free to talk. For a while I was really strict about this. I'm not so strict now. But people really feel that they need a good excuse to come all the way up here, so it's working. I cannot be available for 120 people all the time. That's impossible."

"Has this change made an impact on your knowledge of what is going on at Simula?"

"Yes. Four or five years ago I knew everyone and really felt what the atmosphere was. I knew what was going on all the time. I've lost that. But also, I've got two young kids now, and so I work less. At that time I worked perhaps 60 or 70 hours a week, but now I work about 45. So I have to be stricter on what I am doing, and of course I have lost something. You cannot know everything."

"Also, with 120 people, even if you did have your office in the centre it would be hard to know everybody. Will Simula continue to grow, or is there a right size where it will stabilize?"

"I think it will continue to grow for a while. I think it is a natural tendency for a lab to grow, because the mechanism is that new people enter the lab, they have new ideas, and they want to do things. In order to do new things, you have to hire new people. It's not like in pure math, where many papers are written by just one author. I don't think you'll find a single-authored paper at Simula. There's always a group doing something. The students I had 15 years ago are in a position now where they want to do things on their own. They don't just want to work together with me; they want to be their own project leader. That means growth.

"If you look at the figures on how Simula has developed, you notice that the fraction of non-scientific or support staff has been reduced from 23 per cent to 17 per cent from the start to now. So we are getting more efficient. If you look at the number of permanently hired research scientists, that fraction has also gone down. That means it is hard to get a permanent position here. I think that is good. The number of PhD students has increased very much, the number of postdocs has increased, and also the number of people working with applications has increased.

"I don't know the asymptote on Simula's size, but at least for as long as I can imagine we will stick to our three core subject areas–networks, scientific computing, and software engineering."

"Hiring is obviously an extremely important part of building an organization. Could you talk to me about your general strategies for hiring?"

"In the universities, at least the ones in Norway and Sweden that I know very well, the procedure for hiring a professor is extremely time-consuming. You have to write documents that have to be agreed upon by the department. There is a lot of preparation to post a position, and then you have to open it up for anyone to apply, and often you have quite a large number of applicants. Then there are scientific committees that go through all of the applications and write about all the candidates. This process often takes two years.

"At Simula we wanted to do it differently. We wanted to be fast, and we wanted to do it much more like an ordinary company. The idea in the Norwegian and Swedish universities is very good and very fair, but the practice is poor. We really wanted to do this much more efficiently, looking for really good people and trying to go for them."

"Two of the people I have interviewed for this book, Lionel Briand and Kirsten ten Tusscher, seem to be perfect examples. Can you tell me what attracted you about Lionel and Kirsten?"

"Lionel spent a year here and I talked with him from time to time. He was very enthusiastic about what he was doing and he had very clear thoughts about what to do. When we started talking about a permanent position, I had more of a formal interview with him. We talked for two hours, and he talked for 1 hour and 50 minutes of those two hours. His main topic was how to run Simula! Of course that was a bit unusual for a job interview, but he impressed me very much. I felt we needed that kind of person here. He is very enthusiastic and good with his students, and he has good results coming. He is into deep problems but also problems of relevance for the information technology industry.

"Kirsten is a completely different story. She was invited here by one of our employees to give a talk. I listened to her talk and I enjoyed it very much. I had heard about her earlier and I knew that her papers were well-regarded. At dinner we discussed her job situation, and it turned out that she had a contract that would end in a little bit more than half a year from that time. I asked if she would be interested in coming to Simula, and after a month or so we agreed that she would start here as a research scientist with a little group around her."

"It seems to me, especially in that second case, that the speed with which you were able to act was crucial."

"Yes, absolutely. But I think that was also true for Lionel as well. I think he enjoyed the speed of the process, and the fact that we really went after him."

"Is it difficult to persuade people to leave academia and leave the safety of tenure?"

"We do have tenure, in the sense that people who finish their postdoc can be considered for a permanent position. They are employed permanently in the sense that they have a job as long as Simula will exist. Of course, it's not the degree of certainty you have at a 200-year-old university. Some people enjoy that kind of security, but some enjoy the way we have of working at Simula. There are so few places, at least in Norway, that you can do only research, without having to get money or supervise or teach students. That, of course, is very appealing to many scientists."

"Getting back to the three areas of concentration of Simula, why were those three areas chosen?"

"There was a process arranged by the Research Council, where you could apply to be part of Simula when it was started. That was a national contest. There were 12 applicants, and three groups were selected. None of them were particularly good at the time. That's the truth–you can write that. But in eight years they have become very good. They are at an international level, producing science in international journals at a good rate.

"If you look at these three subjects, you can trace them all back 50 years. Software engineering is about how you program computers to solve large, challenging problems, and design programs that you can renew and that are robust. That has been a problem since the beginning of the computer age, how to create reliable software. It's a very important subject because the entire society depends on software today. If you make a car, 40 per cent of the cost is related to software.

"Of course, also if you go back at least to the 1960s, you will see that communication between computers started very early. That's still a big challenge. Mobile networks break down all the time, but people depend on them, especially in Norway because mobile phones exploded very early here. So we are completely dependent on networks, and the robustness of them is an important problem and has been important for at least 40 years.

"Scientific computing is the same, a very classical problem. The computer was really invented for solving problems in scientific computing. Scientific computing today is what mathematics was 100 years ago. Mathematics was about solving problems and computing solutions. Now scientific computing is about computing solutions."

"That's an interesting point. I noticed that in one of the papers you sent me, there was actually a little proof. I thought, wow, here's a computer guy, and he's still doing proofs! Just asking you to speak as a mathematician, do you think it's a good thing that we have become so dependent on computers? Can it be overdone?"

"That is a question I was not expecting! Actually, my PhD is on proving theorems in hyperbolic conservation laws, proving stability and existence of solutions in nonlinear partial differential equations. That's really what I was trained to do.

"Of course, the frustrating thing about proving things is that you can only address simple problems. When you start digging into the literature about the heart, you see that the more applied the problem is, the more real the problem is, the fewer mathematicians are involved. In the really very challenging problems in that field, you see almost no mathematicians at all. Why is that? I think it is because the tools you traditionally have in math are not really adjusted to deal with those problems. But those problems are extremely important. The only way to relate to them is to use the computer.

"I think my ideal is that you somehow try to figure out a much smaller problem on a simpler geometry and try to understand what is really going on in this equation, from a mathematical point of view. Then you try to complexify it, step by step. You lose the theorems and the estimates, but somehow you keep track of the properties of the problem. Then when you enter a really complex problem and you see that structure, you have understood something even though you don't have the estimates or the theorem that is valid on that geometry. If I have a scientific program for my own research, it's like that. I try to understand the simplest possible problems. Simplify, simplify, simplify. Then I go back the other way and try to keep track of the structure. It's possible!

"When I went to high school, I read about the early 1900s, when there was a revolution in physics. That was when physicists first understood the atom and the nucleus and electrons and how these things work. I thought it must have been extremely exciting to live and to work at that time.

"Many years later I understood that the present period is so much more exciting, and it's because of the computer. It's a fantastic tool to understand physics, and to really solve equations that have been known for 200 or 300 years. For example, in the Navier-Stokes equation, it's only for the last ten years that you have been able to do realistic computations. If you go back to the mid-1980s, the typical geometry would be the unit sphere, the typical number of nodes would be on the order of 1000, and the typical model that you could solve would be a Laplace equation or a heat equation. Now you can do fully realistic geometries and fully realistic models. You can really address the problems that scientists have been aware of for centuries, but have been totally unable to attack in a rigorous manner. That's new! That ability is only ten years old."

"What would Newton think? I think he would have loved this."

"Or von Neumann! I think they would have been thinking about biology. That is the most complex area of computational mathematics.

"If you look at this book (*Mathematical Physiology*) by Jim Keener, who is one of the few mathematicians who has entered this field, you will find a couple of hundred partial differential equations. We are really unable to solve all of them. If you read the preface of that book, Keener says that it would be completely impossible to think of studying astrophysics without doing math. But you can be a medical doctor and do things in physiology without doing math. He thinks that this is completely wrong. You have to do modeling of the parts of the body, of the organs and so on, and you have to format this in the only language available, and that is mathematics.

"The models, both the mathematics and the physics involved in them, are so complex that computers are the only way into these problems. People in these fields realize that."

"How do doctors or physicians respond to these models?"

"Many people we talk to are open-minded, because they are so interested in progress. They don't care so much how the progress is created as long as it is really progress. Many of them see that there is potential in using computers.

"There must be a thousand papers in the field of cardiac arrhythmias and sudden death now, and all of these papers are based on partial differential equations. So there is a strong belief that these equations model these phenomena in an appropriate manner. That is really not questioned. The models are more or less realistic and more or less accurate. We can start to see what really triggers these arrhythmias. That was a bit amazing for me, going into that field from a very theoretical perspective on partial differential equations. I quickly realized that no one would be interested in proofs of existence or things like that. They are only interested in what this equation tells you. What is the content here, what are the properties? All the beautiful theories and all the maximum principles are not interesting at all.

"It's not an easy field for a mathematician to enter, because there are all these funny words that you've never heard before. So I made a living solving these equations in that field for several years before I tried to dig into what the equations were about. I worked on numerical methods to solve them, without really caring too much about the applications. But now, more and more, we are really trying to understand what is going on."

"Hiring people like Kirsten, who is actually a biologist, must be a step in that direction."

"That's not an accident. Absolutely, we want to move in that direction, while keeping a firm grip on modern techniques to solve these equations. That can be our contribution, because we can solve these equations better than the people in the field."

"Changing subjects a little bit, how does Simula inform the public about its work?"

"I don't think we are doing a great job of informing the public about what we are doing, but we are doing a very good job of informing politicians. We have had about 50 members of parliament or the government visiting Simula. When we invite politicians to see Simula, they always accept the invitation. We have one or two scientists telling about what they are doing. The politicians listen and ask good questions, and we have a good dialogue with them. I always have the feeling that they listen to us and understand us. We don't always get what we want, but when we don't get it, I have the feeling that they have at least understood what I'm asking for and why I'm asking for it. I feel we've had a good dialogue, and that's important because this is an enterprise that is different from anything else in Norway.

"Of course there has been a fair amount of Simula in the media. We always try to talk to journalists who want to ask about something, answer them properly and give them the information that they ask for. We haven't spent much money on it, but I think we have been friendly towards anyone asking questions."

"Last question: Where do you see Simula going in the next five years?"

"Morten Dæhlen said that he thought Simula could do very much better on ICT and politics[2]. I totally disagree; that is exactly what we are not going to do. We are going to do much better on figuring out how networks are working, how we do scientific computing and how we develop software. That's really the focus. I think we are on the right track in those directions. I think the Simula School of Research and Innovation will grow and become much better. With Simula Innovation, we have started to understand a little bit more about how to create new businesses. We haven't been successful with that yet, and I'm not satisfied with that part. But we have started to understand more that we really need to focus on these projects.

[2] See the interview with Dæhlen on page 524.

“I think when you come back in five years, you will see that we are in the same fields but we are much better positioned in each field. I hope we don’t spread out; I hope we don’t do ICT politics; and I hope that we still try to think constantly about it.”

3

THE SIMULA CULTURE — HOW WE DO RESEARCH, EDUCATION AND INNOVATION

Aslak Tveito and Marianne M. Sundet

This article describes the culture at Simula; not necessarily as it actually is, but how we would like it to be. It is an attempt to set out some of Simula's defining characteristics.

Describing what is meant by a company culture is no easy task. What are the key characteristics of Simula? What is it that sets us apart? What does a good work atmosphere mean, and how should things be done here to achieve that? An overall, abstract description of workplace culture is likely to be difficult to understand and to relate to daily work, and will also probably be rather similar from one company to the next. That is why we give concrete examples here to make this all more tangible. These examples are not more important than other aspects of the work environment we could have chosen, but have been used merely to make this easy to understand and easier to relate to.

The work of creating a Simula culture involves developing a research laboratory in which a strong focus on high-calibre research is key. Simula will only be successful if its employees are successful. Developing a Simula culture is about creating a work environment where people are happy and feel they are able to do a good job. It is about establishing ground rules for how we act in the research community and interact with society in general, and it is about working to ensure that the name "Simula" is synonymous with quality, honesty and efficiency.

Aslak Tveito · Marianne M. Sundet
Simula Research Laboratory

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_3,

It is important that we all share the same ideas about what sort of culture we want to have, what the culture means and how it can be sustained in the future. Members of staff who have been at Simula for a number of years, some even since its establishment in 2001, will recognise the ideas in the article and hopefully find that the descriptions comply with their own experience. However, Simula has grown and is still growing, and the number of employees is rising. For us, it is paramount that all newcomers, and especially research fellows who are in temporary positions, get the feel of how we relate to each other, our tasks and the world around us. By reading and relating to this article, we hope all employees will understand and find guidance as to how things are done at Simula. For other readers, we hope the article can provide ideas for improvements or at least be a source of thought and discussion about how to organise and run a research lab.

The Simula model

Before we describe the Simula culture, it may be useful to go over the main features of the Simula model[1]:

1. **Organisational model.** The organisational and management models used in the Norwegian research system can basically be divided into the university model and the company model. Research institutes in Norway employ the latter. We strongly favoured the company model, as it appeared to be much clearer, more transparent, and easier to comprehend. Simula is thus managed more or less like a commercial company, with the prominent exception that commercial companies are set up to produce revenues, whereas Simula is constructed to produce research results, educate researchers, and enable innovations. Specifically, Simula is managed according to the Limited Company Act.
2. **Full-time researchers.** At Norwegian universities the teaching and supervising duties are rather severe. On the other hand, at Norwegian research institutes most researchers have to spend a large part of their time trying to obtain funding from industry. Based on the financial freedom that came with Simula, we wanted to establish a situation where highly skilled researchers were allowed to focus more or less exclusively on research; we wanted to revive the full-time researcher.
3. **The recruitment process.** A research lab is as good as the researchers employed there. Therefore it is impossible to exaggerate the importance of a sound recruitment policy. As a part of that, efficiency is crucial; we need to be able to move forward very quickly when a unique opportunity arises. At Simula, we try to recruit extremely promising candidates and the best researchers will be given funds to build their own activity. The recruitment process aims at setting up a truly international lab and whenever possible we try to achieve gender balance.

[1] This description of the Simula model was originally written for the Annual Report 2008.

4. **Directed research.**[2] We never subscribed to the view that excellent research can only be performed when each individual researcher is completely free to follow his or hers individual ideas. Rather, we wanted a model where we could determine a set of long-term goals and collectively work towards these in an organised manner. The research at Simula addresses fundamental problems, but the research is directed more or less as if the activity were organised in a private company. The freedom in day-to-day or week-to-week assignments is very great but the long-term goals are decided in a comprehensive process and everyone has to adapt to these goals. In particular, we have derived a careful procedure of initiating new projects.
5. **Usefulness.** The aim of basic research is rarely to be of use in the short term; indeed, it is widely acknowledged that deep knowledge ultimately is useful but that the path from science to application may be very long. At Simula we address research questions whose solutions would be applicable; that is, we do not pursue strictly curiosity-driven projects. Generally speaking, we address problems where the likelihood for important applications of a positive result is high.
6. **Concentration.** Diffusion is a natural process that moves a substrate from an area of high concentration to a region of low concentration. The process is extremely strong and affects many parts of life. Given a substantial amount of money, there are very strong mechanisms trying to spread these resources over a large number of worthy assignments. At Simula we have tried very hard to maintain focus on core issues and avoid the diffusion of resources.
7. **Collaboration with industry.** Since we want to deliver applicable results, we seek strong collaborations with industry at large. Such collaborations, however, must be long-term and directed towards really challenging problems; short-term consulting should be completely avoided at Simula. Furthermore, our aim is to educate PhDs and postdoctoral candidates with a firm grasp of the problems of industrial interest in their field of research.
8. **Creating new businesses.** Since real-life applications are the long-term goal for our research, Simula consistently aims at aiding researchers to enable the application of their research efforts.
9. **A characteristic culture.** We wanted to create a strong and characteristic research culture based on a few governing principles accepted by everyone, enabling efficiency, quality, excellent results, and a very good working atmosphere. The present version of Simula is based on these building blocks. Of course, none of these parts are unique to Simula, but at least in Norway their combination has apparently not been tried before. Clearly, some of these elements are rather ambitious and we do not claim to have reached good solutions at every point.

[2] In OECD terms, our research can be classified either as *oriented basic research*, which is defined to be research carried out with the expectation that it will produce a broad base of knowledge likely to form the background to the solution of recognised or expected current or future problems or possibilities, or as *applied research*, which is defined to be an original investigation undertaken in order to acquire new knowledge directed primarily towards a specific practical aim or objective (see http://stats.oecd.org/glossary).

Note, however, that creating Simula was founded on very strong ambitions and the determination to create an excellent lab.

How did it all begin?

We were assigned a task and receive State funding to fulfil that task. The task Simula has been set is to carry out research of a high international calibre, to educate students at MSc and PhD level in collaboration with Norwegian universities, and to set up business activities based on the work done at the centre. Simula's research activities fall within three subject areas: networks and distributed systems, scientific computing and software engineering. These are also the three subject areas in which we help to educate students. Readers interested in finding out more about the scientific work carried out at Simula may consult chapters 13, 18, and 24 for networks and distributed systems, scientific computing and software engineering, respectively.

Focus

The area of ambitions equals the width multiplied by the height. All else being equal, the area of ambitions is constant; we may have broad ambitions that are not particularly high, or we may have narrow ambitions that are extremely high, but we must never believe that we can have high ambitions across a wide range of fields. In this respect, Simula has made its choice: we will have high ambitions in a few select fields. That is why the three subject areas we have chosen will remain constant, and we will not spread ourselves too thinly even within these fields. All our research efforts will be devoted to these few carefully chosen areas. But in these areas we will make every effort to succeed. We will participate at the international elite level; we will be invited to speak at professional events; we will be sought-after partners; we will be a natural destination for visiting researchers the world over; we will educate good PhD students; we will be attractive partners for Norwegian industry, and we will develop companies based on our research in these areas. If we are to achieve all of this, we must concentrate our efforts and resources.

The full-time researcher

Wherever researchers meet and discuss their work situations there is one issue that comes up time and time again; researchers find that they simply do not have enough time to devote to research itself. There are a number of reasons for this; they have to apply for research funding to secure the resources needed to run their research groups, they have to teach, they have to serve on committees, they have to mark papers and theses, and they have to support the work of the research councils. A lot of experienced researchers find that their entire working days are spent on activities that are related to research, but that are not research itself.

At Simula, we will make every effort to shield skilled researchers from other duties. We will work hard to revive the concept of the full-time researcher. This is an ambition that is extremely difficult to achieve. We cannot redesign the realities of research policy. All over the world, researchers have always had to work to secure research funding. However, it is widely believed that all these research-related activities are taking up too much time and getting far too much attention. At Simula, we will work constantly and deliberately to reduce the administrative burdens placed on researchers. We want ours to be an efficient organisation. We will not be overly bureaucratic[3]. We will give rapid, clear answers. We will have short meetings. We will have few committees and those that we have will be small. We will not write reports that no one will read. We will not ask for information no one needs. We will know what we want and what our aims are, and we will work with determination to reach those goals. In everything we do, we will be conscious of the fact that it is the scientific staff that is our productive force. They are the ones who produce the results that are the lifeblood of Simula. They are the justification for our existence. That is why we will all strive to ensure the best possible conditions for them to work under.

Research fellows

A large proportion of our researchers are research fellows. Both doctoral and postdoctoral research fellows are employed by Simula. Research fellows are an essential research resource and should be integrated as well as possible into the planning and execution of projects. As far as possible, they should participate on an equal footing throughout the whole research process. Just as a lack of time to conduct research is a recurring topic in discussions among many experienced researchers, the absence of genuine academic supervision is a recurring theme in discussions among research fellows. Research fellows need to receive in-depth feedback on what they are doing. It is both frustrating and inefficient for research fellows to be left to their own devices without feedback about their work. At Simula, supervisors will be accessible to their research fellows electronically, physically and, not least, as mentors. Supervisors must ensure that the research fellows' projects are central to their own research so that supervision is not seen as an "extra", that comes in addition to research. As far as possible, supervision should be research. Further discussion of the role of young researchers is presented in chapter 34 on page 503.

IT Operations and administration

Excellent research needs excellent support. It is the responsibility of the IT Department to ensure that the IT infrastructure supports the research activities as effectively as possible. Operational decisions should always be taken with a view to finding solutions that enhance the focus on research. Solutions that make great demands on

[3] The relation between scientific staff and support staff is discussed on page 74. It is worth noting that the relative number of support staff has declined and the relative number of scientific staff has increased as Simula has expanded from some 30 to some 120 employees.

researchers' time should be avoided as far as possible. We need to use solutions that are stable and that keep operating problems to a minimum. We will try to avoid exposing researchers to unstable solutions from second-rate providers. In this context, quality and stability are, within reason, more important than price. Today, researchers are more or less unable to do anything useful if their PCs are not working. Reliable back-up solutions to all common problems[4] must be available at all times.

The Simula administration provides support for the researchers at the centre and it is this support function that should inform any decisions that are taken. It is the task of the administration to ensure that the research can be carried out under optimal conditions. Procedures should be kept simple, and efforts should be made not to waste other peoples' time unnecessarily. All employees must be aware of and must respect the relevant procedures. The Intranet is an important source of information and must be used actively.

Management

Simula is a limited company. The Board of Directors employs a managing director. He or she takes on directors responsible for Simula's three main units: basic research, research application and research education. Together they constitute the corporate management. The units are defined areas with clear lines of responsibility and communication. The unit directors employ heads of department, directors of subsidiaries, group leaders and other employees. There are no elected positions at any level at Simula. In this respect, Simula differs from universities as we know them today.

The corporate management of Simula, heads of the departments, subsidiaries and of the groups, should always bear in mind that Simula employs extremely well-educated and highly gifted individuals. Strategic issues should be discussed throughout the organisation and any employee who wishes to should have the opportunity to express his or her opinions at all levels. All leaders should be highly accessible and open to discussion about everything that goes on at the centre. All leaders should encourage employees to express their opinions and be prepared to discuss them thoroughly, so that all aspects of an issue are considered before a final decision is made by a leader.

This mode of operation presents great challenges to all leaders at Simula, and it is now fully acknowledged that we have to invest much more in the process of educating our leaders. From 2009, Simula will have its own programme for research leaders.

Directed basic research

At Simula everybody works on research projects that involve other people. The research addresses fundamental problems of an applied nature. The research activities

[4] PCs are stolen, they break down, they are left at airports; mobile phones are also lost and—believe it or not—flushed down the toilet in airplanes. Our IT support staff has to be able to come up with back-up solutions at short notice.

are managed along the same lines as the centre itself. In carrying out the research, it is especially important that all the researchers take part continuously in discussions about how the research should proceed. But here too, decisions are ultimately taken by a leader, who will be accountable for his or her decision. Within the parameters that are set and the research projects that are defined, Simula will strive to ensure that each individual researcher has the freedom to work in the most effective way possible.

Freedom for individual researchers to pursue their own ideas is a key feature of Norwegian universities. There is less freedom to do that at Simula. Everyone has the opportunity to take part fully in discussions about the choice of strategy, working methods and research tasks. However, once those decisions are made, everyone must work to achieve the targets set for the research group. The researchers will have the greatest possible freedom to make their own choices within the parameters set, but they cannot choose their own areas of research independent of the constraints chosen for the project as a whole. In other words, the freedom of individual researchers at Simula is limited compared with similar positions at universities.

At Simula we carry out directed basic research. We set targets for our activities and then work hard to achieve them. This places restrictions on the freedom of the individual. However, the freedom of a group is considerable, the freedom of a department is even greater and the freedom enjoyed by Simula is extremely large. But, we must have a common understanding of what we are trying to achieve, the problems we intend to solve and how we intend to solve them.

Although the freedom of individual researchers to pursue their own ideas at Simula is restricted, Simula is an organisation characterised by openness and intense debates. Traditional academic ideals regarding free debate and vigorous discussion are a key part of the Simula culture.

Freedom, responsibility and initiative

At Simula, freedom divided by responsibility is constant. A large amount of freedom carries with it a large amount of responsibility. All the researchers and all the research groups have considerable freedom, but this freedom is accompanied by responsibility; the responsibility to conduct valuable research and deliver valuable results, and to carry out that research in accordance with the Simula culture. Simula will welcome good initiatives, but they need to be followed up and developed. At Simula, employees must be able to support their ideas and be prepared to work hard to put into place any necessary measures.

Publications

Simula aims to publish its scientific results in the leading professional journals, in books and/or in good conference proceedings. The research will be focused on issues that are important and relevant. It will not be focused on the most publishable issues. Researchers at Simula will publish when an important problem has been solved or new insight has been gained, and will not publish their findings simply to boost the number of publications. Important publications will be given clear priority over less significant observations. At international gatherings of researchers, our researchers will take the floor when they have something new to tell the research community.

Publishing research findings is vital for researchers trying to establish their careers. Our record of publications is used when we apply for new positions, when we apply for professorships, for funding from the Research Council of Norway, etc. So it is natural that younger researchers are very keen to publish their work as often as possible. Also, learning to write scholarly articles is an important part of a PhD. Younger researchers need to learn to write articles even when the results are not ground-breaking. Simula accepts that this as an essential part of career development for individual researchers. We also accept that evaluations of Simula will, to some extent, be based on the number of publications we produce. But we will not lose sight of our ideal; to provide new insight into important issues of value to others.

Quality

Quality is paramount at Simula. Quality should be all-important in everything we do, everything we deliver, in our whole working day. This means that:

- All our publications will go through a thorough quality control.
- Everyone at Simula is free to express their opinions in the media, preferably in the form of letters to the editor, feature articles or similar contributions. But if an author signs any piece of work as a Simula employee, it must be of a high quality; careless work, tactless remarks etc. must be avoided.
- Everything Simula buys (all the software, machinery, equipment, coffee cups, stationary, Christmas gifts, T-shirts, etc.) must be of a high quality. We will not buy unnecessarily expensive products, but we will *never* buy low-quality products, and we will always have a view towards making an environmentally friendly choice.
- Simula employees should be on time for appointments and should meet agreed deadlines. If for some reason this is not possible, the relevant person should be notified regardless of whether or not an appointment is especially important. When we meet others in connection with work, we represent Simula, which means that we have a reputation to maintain. A high level of professionalism is a key component of that reputation. Attention should be paid to the nature of the meeting and participants should dress accordingly.
- We will give a warm welcome to any visitors we receive. Meetings should be well prepared. All technical equipment to be used at a meeting should be checked in advance. Overhead transparencies should be printed out on paper or sent as attachments. Meeting rooms, coffee/tea and refreshments should be booked and ordered in advance. It is important to make sure that everything runs smoothly at all meetings so that this forms part of the overall impression a visitor to Simula receives. Simula will receive new graduate students in the same way, showing them respect and treating them as equals.

- All new employees at Simula should be received by their immediate manager. The head of department or director of the subsidiary are responsible for ensuring that they are introduced to colleagues and shown around the premises, and that they are acquainted with procedures, standards and rules, as well as their rights as Simula employees. During the first week, new employees will make their own home page and have their photograph taken. An email containing a brief introduction to the employee in question will then be sent out, along with a link to the new employee's home page.
- Presentations of our activities should be well prepared. All employees should be able to introduce Simula and give an overview of what we do.
- Anyone representing Simula should always behave professionally. Telephone calls should be answered in a friendly manner, including telephone calls from people trying to sell us something. Emails from Simula that are not of a personal nature or not to people we know well, are to be written in correct language. The sender's address should be @simula.no; the use of Hotmail, Gmail accounts etc. should be avoided in professional emails from Simula. Simula, including the subsidiaries Simula Innovation and Simula School of Research and Innovation, is subject to the law on the right of access to public administration documents (Freedom of Information Act). The act stipulates that the organisation's documents should be made available to the public, and we are obliged to comply with a system for managing documents. To ensure that documents reach the archives and are assessed with regard to the public record it is necessary that everyone complies with the procedures for managing documents and post.
- As Simula employees, we should respond to queries quickly and succinctly. If in doubt, ask your immediate manager for help.
- When Simula employees are given an assignment or are asked to do something, they should always send a reply so that the sender knows that the assignment has been received and when it will be done.

We should be on the look out for any major and minor improvements we can make. Is this something Simula as a whole could do better? Are there areas in which we have poor procedures or in which we come across as disorganised or unprofessional? Are there any mistakes on our website? Is the reception area untidy? Does all our technical equipment work properly? Everyone should be on the look out for such matters and point them out to the person they believe to be responsible. We can correct little things ourselves, but larger issues should be reported to an immediate manager. And such queries should be followed up. A few days later, ask what happened. What has been done? Was anything done? There will, of course, be things we choose to do nothing about, but there should be a good reason for that and the person who brought up the issue should receive a clear explanation as to why that is the case.

Integrity

Simula makes its living from the truth. Our job is to uncover connections and convey them to other scientists and to the public at large. In all research communities, this calls for reliability and honesty. Simula will go further than this. Integrity will be paramount in everything we do. We will act fairly towards students, partners, vendors, the authorities, the media and each other. We will give praise and recognition based on merit and not on position. Whenever the Simula name is used, the information given should be reliable and accurate. Our web pages will include up-to-date, accurate information. Our annual reports will be accurate. Our surveys of publications, patents, enterprises, etc. will be truthful and always completely verifiable. Simula will never "tweak" numbers or reports to make them look better than they really are; nor will Simula ever present its research as better, or more important, than it really is.

Credibility

Simula's reputation depends on its credibility with the Norwegian IT industry, the authorities and other partners. Credibility demands that we behave professionally with regard to agreements and deliveries; we anonymise results from industrial studies, unless otherwise agreed; we demonstrate considerable caution in what we say and write about our partners in presentations and at meetings; and we maintain a continuous focus on good research ethics in our studies.

Efficiency

Simula should be an efficient organisation. All queries directed to Simula should receive a response within one working day. Naturally, it will sometimes be necessary to reply that a full answer may take time, but where appropriate, queries to a Simula employee should receive a response within one working day.

At Simula, it is assumed that departments, research groups, subsidiaries and the individual employees will work in accordance with the plans that are drawn up. To carry out this work, the department, research group, subsidiaries and individual employee will have as much freedom and as much responsibility as possible. The aim is to promote efficiency. Decisions are not to be left hanging. Any employee can bypass his or her immediate manager and go to the next level of authority if the manager in question fails to make a decision within a reasonable length of time. However, this should not be done lightly and should only be done in cases where an employee finds that his or her immediate manager is too slow, or where his or her decision is obviously inappropriate or motivated by personal interests.

The media

The Government pays our salaries because it believes that research is valuable. Our work is valuable because it enhances insight into complex problems, contributes to the education of qualified people and leads to new solutions. However, research is also important because researchers share their findings with society through the media, either through their own contributions or through interviews. For that reason it is important that researchers at Simula make themselves available to the media when appropriate, and that researchers contact the media when they have significant scientific findings to communicate. As mentioned above, all Simula employees are free to express their opinions in the media. However, when doing so, Simula employees should be clear about whether they are giving an opinion as a Simula employee or as a private individual. When making a statement as a Simula employee, it is important to be aware that Simula's reputation is at stake. Careless work and inaccuracy must be avoided. Furthermore, employees of Simula should be careful not to come across as experts on a topic they are not actually researching themselves. In cases concerning Simula as an institution, only the Managing Director will respond to questions from the media.

Integration

All PhD and postdoctoral positions at Simula are announced internationally. To a large extent Simula recruits employees, both from international and national academic environments, through headhunting. At Simula we will do our best to make sure that everyone finds their way around when they are new to the country or the area, both in the workplace and socially. We will make sure that new colleagues are introduced to the workplace. Colleagues within the departments and groups should also make an effort to show newcomers around and help them get started with the process of finding friends and a place to live, and cope with the practical issues associated with relocation. In 2008, a survey was conducted for the purpose of assessing whether employees from abroad were properly welcomed. The survey showed that, in general, our policy is working well, but there are still some specific areas where there is scope for improvement.

Equality and diversity

Simula should be a good place to work. It should be a place where all employees feel that their jobs are important, where they understand why they do their jobs and why their jobs are important for Simula. At Simula there is no room for condescending attitudes to colleagues or to other vocational groups. Embracing diversity enhances an organisation's range of experience, ideas and creativity. Simula will encourage all employees to recognise and respect the diversity of their colleagues in terms of gender, age, sexual orientation or national origin. All employees and all students are important and will, without exception, be treated as such.

The workplace

Simula should be a good place to work when everything is going well; it should be a place where employees are able to realise their ambitions. But Simula should also be a good place to work when things are not going so well, whether on the professional or the personal front. Consideration for colleagues and recognition that all employees are individuals with individual needs should be dominant features of the work environment. Whether an employee has just had a baby or is ready to retire, Simula should be a good place to work, just as it should be for a 30-year-old researcher hungry to conquer the world of science. Simula will work to create a work environment where people are happy and one that is stimulating and creative. However, such a working environment does not develop by itself. Everyone at Simula should feel a sense of shared responsibility for creating a good working environment. Employees at Simula should appreciate each other's efforts across disciplinary divides. Everyone must accept and respect the fact that Simula will thrive when everyone thrives, and for that to happen, everyone must feel that their efforts are valued by the lab as a whole.

4

IMPRESSED WITH TARGETED RESEARCH STRATEGY

An interview with Arvid Hallén by Bjarne Røsjø

"One of the things about Simula that has impressed me is that they have managed to combine a relatively healthy financial situation with a highly targeted approach to managing research activities," says Arvid Hallén, Director General of the Research Council of Norway.

Having been head of the Research Council since 2004, Arvid Hallén is well acquainted with Simula. The same year, an international evaluation committee, appointed by the Research Council, delivered an extremely positive evaluation of Simula. Every year the Research Council provides basic funding of about 50 million NOK to Simula based on allocations from the Ministry of Education and Research, the Ministry of Trade and Industry and the Ministry of Transport and Communications. In addition, the Research Council provides funding for the Center for Biomedical Computing, which is a Centre of Excellence at Simula. It also contributes to individual research projects. A former head of the Norwegian Institute for Urban and Regional Research (NIBR), Hallén joined the Research Council in 1995 as Executive Director of the Division for Culture and Society. In 2003 he was made Assistant Executive Director of the Research Council's Division for Science.

Discontinuing research takes courage

"Simula has taken a targeted approach to research in the sense that they have consistently proceeded with projects that have shown promise, but they have also discontinued or redefined projects that have not shown such potential. Simula has managed to implement a targeted approach to managing its research activities, driven by the needs of the research itself, and has maintained a sufficiently stable financial situation

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_4,

to be able to chart its own course. This has proven to be a very effective combination. It takes courage to abandon projects that are not going very well," Hallén points out. He believes that it is precisely this targeted research strategy that can largely account for Simula's achieving such good results so soon after its establishment in 2001.

"Within the context of the Norwegian research system Simula is unique. Given that the establishment of Simula has been such a success, does the Research Council envisage that more centres could be established along the lines of Simula?"

"Yes, the Simula model is somewhat unique. The first evaluation of Simula was extremely positive and the centre is now due to be evaluated again. This time we will also be carrying out an evaluation of Simula as a concept. The purpose of this is to obtain a thorough and independent assessment of how far this model could be applied in other areas," he replies.

Bold initiatives

Hallén adds that over the past few decades in Norway a long time has passed between each time a completely new research institution has been established. "Now we are strengthening our commitment to new areas by making use of the national competitive arenas and the institutions that already exist. The various centre schemes are examples of this, the most recent initiative being the establishment of the new Centres for Environmentally-friendly Energy Research (CEER). Here the institutions compete, and funding is allocated to the research communities that can demonstrate solid scientific achievements and that, in addition, draw up the best research plans. The establishment of Simula, on the other hand, was more reminiscent of Norwegian research policy in the decades following World War II. Then bold initiatives were undertaken to establish new institutions designed to take care of areas considered in need of strengthening," Hallén points out.

Although Simula is unique, its establishment was based on the same reasoning as the Centres of Excellence scheme. The aim of the CoE scheme is to create centres dedicated to long-term basic research of a high international calibre, and to raise the quality of Norwegian research by giving the most competent research communities better and more long-term funding.

"We have now established 21 CoEs. A mid-term evaluation of the first 13 centres revealed that the scheme has so far yielded very positive results. Simula is in an even better position than the CoEs; Simula could almost be described as several CoEs rolled into one. It is clear that this has enabled Simula to enhance its reputation and profile both at the national and the international level," Hallén comments.

Strong focus yields results

Hallén notes that both Simula and the CoE scheme demonstrate that good results come from a targeted and focused strategy. "We know that we need national institutions that are respected and have a high profile internationally, if we are to promote Norway as a research nation in the face of increasingly intense competition. There

Arvid Hallén

is no doubt in my mind that the CoE scheme, Simula and the leading universities will play a key role in our achieving this. But at the Research Council we also have to make sure that we maintain an overall balance between different initiatives in order to obtain the best results from a national standpoint. It is obvious that if you allocate a huge amount of resources to building a centre like Simula, you take away resources from other research communities. This is unfortunate if it undermines the bigger picture."

"Is there any indication that Simula has sapped expertise or other resources from other Norwegian research institutions?"

"That is something we do not know enough about. We have certainly not received definitive reports to that effect. But we do know that Simula has been able to offer researchers very good terms and conditions, probably better than they would have had at a university," Hallén replies.

"These days I think that there is a much broader acceptance in Norway of the need to build up strong research communities, and I have full confidence in the steps we have taken so far. We will most likely continue the policy of strengthening the concentration of resources, and we may establish more leading research centres along the lines of Simula. But there will probably not be scope for a lot of initiatives of that kind," Hallén concludes.

5

THE HAMMING EXPERIENCE

Hans Petter Langtangen and Olav Lysne

Many scientists frequently wonder about the answers to questions, such as *What characterizes high-quality scientific research?*, *How does it differ from that which is more routine in nature?*, and *What steps must I, as a scientist, take to improve myself, so that I might reach the top of my field?*

Very often we recall world-class, renowned scientists or personal mentors whom we revere when we ponder such questions. However, deducing the methods behind the work, the nature of their day-to-day duties and responsibilities, and how they achieved their greatest triumphs can be a difficult proposition. Research is a fuzzy, often nebulous process that can be difficult to describe in words. One such researcher, Richard Hamming, was once asked to give a talk about being a first-class scientist. Fortunately, his words were recorded and transcribed. In his talk, Hamming addressed our questions above and many others of major importance in a very inspiring manner that explains in explicit detail what it takes to do first-class scientific research and which failures are likely to occur. We find Hamming's words so inspiring and useful that we have decided to include them in this book.

One aspect of Hamming's talk is of particular significance to this book. Those who built Simula wanted to create a research laboratory that, over time, could host top scientists and be an inspiration to those who aspired to do first-class research. Gleaning the answers to the questions posed above is thus of the outmost importance. After reading the transcription of Hamming's talk, we found that it–in a remarkable way–distilled our attitudes, thoughts, and insight into what research actually is and what being a researcher is really about. The ideas and messages from his talk have inspired and guided us in the process of building Simula. We want Simula to

Hans Petter Langtangen · Olav Lysne
Simula Research Laboratory

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_5,
© Springer-Verlag Berlin Heidelberg 2010

be an institution that actively implements Hamming's ideas for achieving success in research and avoids the potential for failure.

Hamming was speaking to researchers desirous of doing work of the very highest calibre, the type that can lead to awards comparable to the Nobel prize. While such work is a noble undertaking, we believe the audience for his words should be broadened to include all those who would study science. For example, most of his descriptions and recommendations apply to all kinds of scientific work, including that undertaken by students. This is why we wanted to include his famous talk in this book and let it serve as an inspiration for everyone who appreciates the pleasure of doing high-quality scientific work.

Many of the critical topics for success in science are brought up by Hamming. First, he says, you must drop modesty–you must believe in yourself and explicitly express it, for example, by stating, *Yes, I want to achieve something significant.* To achieve something significant, you must work on an important problem. He asserts that too many scientists work on what, at heart, they believe to be unimportant problems. Therefore, to achieve significant results, you (or your institutions) must discover sufficiently important problems as prerequisites for performing research that stands a chance of making an impact.

Exhibiting a tremendous drive is a necessary condition for a researcher's success. Hamming's formulation is: *Knowledge and productivity are like compound interest. Given two people with exactly the same ability, the one person who manages day in and day out to get in one more hour of thinking will be tremendously more productive over a lifetime.* In addition, if you have a problem for which you seek a solution, you need to be truly committed to your problem, so that your subconscious mind can work on the problem, too. But working hard and being deeply immersed in a topic are not enough–you must also be sensible, both in your approach to the work and in your thinking.

Along with the qualities of sheer confidence, tremendous drive, and a fresh perspective, looking back over their careers, many scientists feel that their achievements occurred as a result of luck. Hamming warns against waiting for the lucky moments, citing Pasteur's words: "Luck favours the prepared mind." This means that the more you know about a subject, the more you think about it; and also, the greater the amount of courage you feel as you pursue your ideas, the more "luck" will come your way. Great achievements also often stem from being able to both believe and doubt a hypothesis at the same time: you must believe it enough to go forward with it, but doubt it enough to realize its deficiencies and from them, devise improvements that eventually could lead to great work.

In the above paragraphs, we have noted a few of Hamming's views. Still, Hamming approaches these questions from the standpoint of a researcher, not from that of an institution. Therefore, in the formation of Simula, we felt a need to address the relationship between the successful researcher and the successful institution. One way of formulating this is *How can we create an environment in which people like Hamming could thrive and be productive?* Simula's answer to this question is formulated by Aslak Tveito and Morten Dæhlen in chapter 1, and is further elaborated in the description of the Simula culture in chapter 3.

The full value of Hamming's thoughts on these matters can only be appreciated by reading the transcript yourself. Many of us who have had the pleasure of participating in Simula's formation have done so, time and time again. If you have not read it before, we can assure you that it is well worth your while.

Enjoy!

6

RICHARD HAMMING — YOU AND YOUR RESEARCH

Transcription of the
Bell Communications Research Colloquium Seminar
7 March 1986

At a seminar in the Bell Communications Research Colloquia Series, Dr. Richard W. Hamming, a Professor at the Naval Postgraduate School in Monterey, California and a retired Bell Labs scientist, gave a very interesting and stimulating talk, *You and Your Research* to an overflow audience of some 200 Bellcore staff members and visitors at the Morris Research and Engineering Center on March 7, 1986. This talk centered on Hamming's observations and research on the question "Why do so few scientists make significant contributions and so many are forgotten in the long run?" From his more than forty years of experience, thirty of which were at Bell Laboratories, he has made a number of direct observations, asked very pointed questions of scientists about what, how, and why they did things, studied the lives of great scientists and great contributions, and has done introspection and studied theories of creativity. The talk is about what he has learned in terms of the properties of the individual scientists, their abilities, traits, working habits, attitudes, and philosophy.

J. F. Kaiser
Bell Communications Research
445 South Street
Morristown, NJ 07962-1910
jfk@bellcore.com

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_6,

In order to make the information in the talk more widely available, the tape recording that was made of that talk was carefully transcribed. This transcription includes the discussions which followed in the question and answer period. As with any talk, the transcribed version suffers from translation as all the inflections of voice and the gestures of the speaker are lost; one must listen to the tape recording to recapture that part of the presentation. While the recording of Richard Hamming's talk was completely intelligible, that of some of the questioner's remarks were not. Where the tape recording was not intelligible I have added in parentheses my impression of the questioner's remarks. Where there was a question and I could identify the questioner, I have checked with each to ensure the accuracy of my interpretation of their remarks.

Introduction of dr. Richard W. Hamming

As a speaker in the Bell Communications Research Colloquium Series, Dr. Richard W. Hamming of the Naval Postgraduate School in Monterey, California, was introduced by Alan G. Chynoweth, Vice President, Applied Research, Bell Communications Research.

Alan G. Chynoweth: Greetings colleagues, and also to many of our former colleagues from Bell Labs who, I understand, are here to be with us today on what I regard as a particularly felicitous occasion. It gives me very great pleasure indeed to introduce to you my old friend and colleague from many many years back, Richard Hamming, or Dick Hamming as he has always been know to all of us.

Dick is one of the all time greats in the mathematics and computer science arenas, as I'm sure the audience here does not need reminding. He received his early education at the Universities of Chicago and Nebraska, and got his PhD at Illinois; he then joined the Los Alamos project during the war. Afterwards, in 1946, he joined Bell Labs. And that is, of course, where I met Dick–when I joined Bell Labs in their physics research organization. In those days, we were in the habit of lunching together as a physics group, and for some reason, this strange fellow from mathematics was always pleased to join us. We were always happy to have him with us because he brought so many unorthodox ideas and views. Those lunches were stimulating, I can assure you.

While our professional paths have not been very close over the years, nevertheless I've always recognized Dick in the halls of Bell Labs and have always had tremendous admiration for what he was doing. I think the record speaks for itself. It is too long to go through all the details, but let me point out, for example, that he has written seven books and of those seven books which tell of various areas of mathematics and computers and coding and information theory, three are already well into their second edition. That is testimony indeed to the prolific output and the stature of Dick Hamming.

I think I last met him–it must have been about ten years ago–at a rather curious little conference in Dublin, Ireland where we were both speakers. As always, he was tremendously entertaining. Just one more example of the provocative thoughts that he comes up with: I remember him saying, “There are wavelengths that people cannot see, there are sounds that people cannot hear, and maybe computers have thoughts that people cannot think.” Well, with Dick Hamming around, we don’t need a computer. I think that we are in for an extremely entertaining talk.

The Talk: “You and Your Research” by Dr. Richard W. Hamming

It’s a pleasure to be here. I doubt if I can live up to the Introduction. The title of my talk is, “You and Your Research.” It is not about managing research, it is about how you individually do your research. I could give a talk on the other subject–but it’s not, it’s about you. I’m not talking about ordinary run-of-the-mill research; I’m talking about great research. And for the sake of describing great research I’ll occasionally say Nobel-Prize type of work. It doesn’t have to gain the Nobel Prize, but I mean those kinds of things which we perceive are significant things. Relativity, if you want, Shannon’s information theory, any number of outstanding theories–that’s the kind of thing I’m talking about.

Now, how did I come to do this study? At Los Alamos I was brought in to run the computing machines which other people had got going, so those scientists and physicists could get back to business. I saw I was a stooge. I saw that although physically I was the same, they were different. And to put the thing bluntly, I was envious. I wanted to know why they were so different from me. I saw Feynman up close. I saw Fermi and Teller. I saw Oppenheimer. I saw Hans Bethe: he was my boss. I saw quite a few very capable people. I became very interested in the difference between those who do and those who might have done.

When I came to Bell Labs, I came into a very productive department. Bode was the department head at the time; Shannon was there, and there were other people. I continued examining the questions, “Why?” and “What is the difference?” I continued subsequently by reading biographies, autobiographies, asking people questions such as: “How did you come to do this?” I tried to find out what are the differences. And that’s what this talk is about.

Now, why is this talk important? I think it is important because, as far as I know, each of you has one life to live. Even if you believe in reincarnation it doesn’t do you any good from one life to the next! Why shouldn’t you do significant things in this one life, however you define significant? I’m not going to define it–you know what I mean. I will talk mainly about science because that is what I have studied. But so far as I know, and I’ve been told by others, much of what I say applies to many fields. Outstanding work is characterized very much the same way in most fields, but I will confine myself to science.

In order to get at you individually, I must talk in the first person. I have to get you to drop modesty and say to yourself, “Yes, I would like to do first-class work.” Our society frowns on people who set out to do really good work. You’re not supposed to; luck is supposed to descend on you and you do great things by chance. Well, that’s a kind of dumb thing to say. I say, why shouldn’t you set out to do something significant. You don't have to tell other people, but shouldn't you say to yourself, "Yes, I would like to do something significant."

In order to get to the second stage, I have to drop modesty and talk in the first person about what I've seen, what I've done, and what I've heard. I'm going to talk about people, some of whom you know, and I trust that when we leave, you won't quote me as saying some of the things I said.

Let me start not logically, but psychologically. I find that the major objection is that people think great science is done by luck. It's all a matter of luck. Well, consider Einstein. Note how many different things he did that were good. Was it all luck? Wasn't it a little too repetitive? Consider Shannon. He didn't do just information theory. Several years before, he did some other good things and some which are still locked up in the security of cryptography. He did many good things.

You see again and again, that it is more than one thing from a good person. Once in a while a person does only one thing in his whole life, and we'll talk about that later, but a lot of times there is repetition. I claim that luck will not cover everything. And I will cite Pasteur who said, "Luck favors the prepared mind." And I think that says it the way I believe it. There is indeed an element of luck, and no, there isn't. The prepared mind sooner or later finds something important and does it. So yes, it is luck. The particular thing you do is luck, but that you do something is not.

For example, when I came to Bell Labs, I shared an office for a while with Shannon. At the same time he was doing information theory, I was doing coding theory. It is suspicious that the two of us did it at the same place and at the same time–it was in the atmosphere. And you can say, "Yes, it was luck." On the other hand you can say, "But why of all the people in Bell Labs then were those the two who did it?" Yes, it is partly luck, and partly it is the prepared mind; but "partly" is the other thing I'm going to talk about. So, although I'll come back several more times to luck, I want to dispose of this matter of luck as being the sole criterion whether you do great work or not. I claim you have some, but not total, control over it. And I will quote, finally, Newton on the matter. Newton said, "If others would think as hard as I did, then they would get similar results."

One of the characteristics you see, and many people have it including great scientists, is that usually when they were young they had independent thoughts and had the courage to pursue them. For example, Einstein, somewhere around 12 or 14, asked himself the question, "What would a light wave look like if I went with the velocity of light to look at it?" Now he knew that electromagnetic theory says you cannot have a stationary local maximum. But if he moved along with the velocity of light, he would see a local maximum. He could see a contradiction at the age of 12, 14, or somewhere around there, that everything was not right and that the velocity of light had something peculiar. Is it luck that he finally created special relativity? Early on, he had laid down some of the pieces by thinking of the fragments. Now that's the necessary but not sufficient condition. All of these items I will talk about are both luck and not luck.

How about having lots of "brains?" It sounds good. Most of you in this room probably have more than enough brains to do first-class work. But great work is something else than mere brains. Brains are measured in various ways. In mathematics, theoretical physics, astrophysics, typically brains correlates to a great extent with the ability to manipulate symbols. And so the typical IQ test is apt to score them fairly high. On the other hand, in other fields it is something different. For example, Bill Pfann, the fellow who did zone melting, came into my office one day. He had this idea dimly in his mind about what he wanted and he had some equations. It was pretty clear to me that this man didn't know much mathematics and he wasn't really articulate. His problem seemed interesting so I took it home and did a little work. I finally showed him how to run computers so he could compute his own answers. I gave him the power to compute. He went ahead, with negligible recognition from his own department, but ultimately he has collected all the prizes in the field. Once he got well started, his shyness, his awkwardness, his inarticulateness, fell away and he became much more productive in many other ways. Certainly he became much more articulate.

And I can cite another person in the same way. I trust he isn't in the audience, i.e. a fellow named Clogston. I met him when I was working on a problem with John Pierce's group and I didn't think he had much. I asked my friends who had been with him at school, "Was he like that in graduate school?" "Yes," they replied. Well I would have fired the fellow, but J. R. Pierce was smart and kept him on. Clogston finally did the Clogston cable. After that there was a steady stream of good ideas. One success brought him confidence and courage.

One of the characteristics of successful scientists is having courage. Once you get your courage up and believe that you can do important problems, then you can. If you think you can't, almost surely you are not going to. Courage is one of the things that Shannon had supremely. You have only to think of his major theorem. He wants to create a method of coding, but he doesn't know what to do so he makes a random code. Then he is stuck. And then he asks the impossible question, "What would the average random code do?" He then proves that the average code is arbitrarily good, and that therefore there must be at least one good code. Who but a man of infinite courage could have dared to think those thoughts? That is the characteristic of great scientists; they have courage. They will go forward under incredible circumstances; they think and continue to think.

Age is another factor which the physicists particularly worry about. They always are saying that you have got to do it when you are young or you will never do it. Einstein did things very early, and all the quantum mechanic fellows were disgustingly young when they did their best work. Most mathematicians, theoretical physicists, and astrophysicists do what we consider their best work when they are young. It is not that they don't do good work in their old age but what we value most is often what they did early. On the other hand, in music, politics and literature, often what we consider their best work was done late. I don't know how whatever field you are in fits this scale, but age has some effect.

But let me say why age seems to have the effect it does. In the first place if you do some good work you will find yourself on all kinds of committees and unable to do any more work. You may find yourself as I saw Brattain when he got a Nobel Prize. The day the prize was announced we all assembled in Arnold Auditorium; all three winners got up and made speeches. The third one, Brattain, practically with tears in his eyes, said, “I know about this Nobel-Prize effect and I am not going to let it affect me; I am going to remain good old Walter Brattain.” Well I said to myself, “That is nice.” But in a few weeks I saw it was affecting him. Now he could only work on great problems.

When you are famous it is hard to work on small problems. This is what did Shannon in. After information theory, what do you do for an encore? The great scientists often make this error. They fail to continue to plant the little acorns from which the mighty oak trees grow. They try to get the big thing right off. And that isn’t the way things go. So that is another reason why you find that when you get early recognition it seems to sterilize you. In fact I will give you my favorite quotation of many years. The Institute for Advanced Study in Princeton, in my opinion, has ruined more good scientists than any institution has created, judged by what they did before they came and judged by what they did after. Not that they weren’t good afterwards, but they were superb before they got there and were only good afterwards.

This brings up the subject, out of order perhaps, of working conditions. What most people think are the best working conditions, are not. Very clearly they are not because people are often most productive when working conditions are bad. One of the better times of the Cambridge Physical Laboratories was when they had practically shacks–they did some of the best physics ever.

I give you a story from my own private life. Early on it became evident to me that Bell Laboratories was not going to give me the conventional acre of programming people to program computing machines in absolute binary. It was clear they weren’t going to. But that was the way everybody did it. I could go to the West Coast and get a job with the airplane companies without any trouble, but the exciting people were at Bell Labs and the fellows out there in the airplane companies were not. I thought for a long while about, “Did I want to go or not?” and I wondered how I could get the best of two possible worlds. I finally said to myself, “Hamming, you think the machines can do practically everything. Why can’t you make them write programs?” What appeared at first to me as a defect forced me into automatic programming very early. What appears to be a fault, often, by a change of viewpoint, turns out to be one of the greatest assets you can have. But you are not likely to think that when you first look the thing and say, “Gee, I’m never going to get enough programmers, so how can I ever do any great programming?”

And there are many other stories of the same kind; Grace Hopper has similar ones. I think that if you look carefully you will see that often the great scientists, by turning the problem around a bit, changed a defect to an asset. For example, many scientists when they found they couldn’t do a problem finally began to study why not. They then turned it around the other way and said, “But of course, this is what it is” and got an important result. So ideal working conditions are very strange. The ones you want aren’t always the best ones for you.

Now for the matter of drive. You observe that most great scientists have tremendous drive. I worked for ten years with John Tukey at Bell Labs. He had tremendous drive. One day about three or four years after I joined, I discovered that John Tukey was slightly younger than I was. John was a genius and I clearly was not. Well I went storming into Bode’s office and said, “How can anybody my age know as much as John Tukey does?” He leaned back in his chair, put his hands behind his head, grinned slightly, and said, “You would be surprised Hamming, how much you would know if you worked as hard as he did that many years.” I simply slunk out of the office!

What Bode was saying was this: “Knowledge and productivity are like compound interest.” Given two people of approximately the same ability and one person who works ten per cent more than the other, the latter will more than twice outproduce the former. The more you know, the more you learn; the more you learn, the more you can do; the more you can do, the more the opportunity–it is very much like compound interest. I don’t want to give you a rate, but it is a very high rate. Given two people with exactly the same ability, the one person who manages day in and day out to get in one more hour of thinking will be tremendously more productive over a lifetime. I took Bode’s remark to heart; I spent a good deal more of my time for some years trying to work a bit harder and I found, in fact, I could get more work done. I don’t like to say it in front of my wife, but I did sort of neglect her sometimes; I needed to study. You have to neglect things if you intend to get what you want done. There’s no question about this.

On this matter of drive Edison says, “Genius is 99% perspiration and 1% inspiration.” He may have been exaggerating, but the idea is that solid work, steadily applied, gets you surprisingly far. The steady application of effort with a little bit more work, *intelligently applied* is what does it. That’s the trouble; drive, misapplied, doesn’t get you anywhere. I’ve often wondered why so many of my good friends at Bell Labs who worked as hard or harder than I did, didn’t have so much to show for it. The misapplication of effort is a very serious matter. Just hard work is not enough–it must be applied sensibly.

There’s another trait on the side which I want to talk about; that trait is ambiguity. It took me a while to discover its importance. Most people like to believe something is or is not true. Great scientists tolerate ambiguity very well. They believe the theory enough to go ahead; they doubt it enough to notice the errors and faults so they can step forward and create the new replacement theory. If you believe too much you’ll never notice the flaws; if you doubt too much you won’t get started. It requires a lovely balance. But most great scientists are well aware of why their theories are true and they are also well aware of some slight misfits which don’t quite fit and they don’t forget it. Darwin writes in his autobiography that he found it necessary to write down every piece of evidence which appeared to contradict his beliefs because otherwise they would disappear from his mind. When you find apparent flaws you’ve got to be sensitive and keep track of those things, and keep an eye out for how they can be explained or how the theory can be changed to fit them. Those are often the great contributions. Great contributions are rarely done by adding another decimal place. It comes down to an emotional commitment. Most great scientists are completely committed to their problem. Those who don’t become committed seldom produce outstanding, first-class work.

Now again, emotional commitment is not enough. It is a necessary condition apparently. And I think I can tell you the reason why. Everybody who has studied creativity is driven finally to saying, “creativity comes out of your subconscious.”

Somehow, suddenly, there it is. It just appears. Well, we know very little about the subconscious; but one thing you are pretty well aware of is that your dreams also come out of your subconscious. And you're aware your dreams are, to a fair extent, a reworking of the experiences of the day. If you are deeply immersed and committed to a topic, day after day after day, your subconscious has nothing to do but work on your problem. And so you wake up one morning, or on some afternoon, and there's the answer. For those who don't get committed to their current problem, the subconscious goofs off on other things and doesn't produce the big result. So the way to manage yourself is that when you have a real important problem you don't let anything else get the center of your attention–you keep your thoughts on the problem. Keep your subconscious starved so it has to work on your problem, so you can sleep peacefully and get the answer in the morning, free.

Now Alan Chynoweth mentioned that I used to eat at the physics table. I had been eating with the mathematicians and I found out that I already knew a fair amount of mathematics; in fact, I wasn't learning much. The physics table was, as he said, an exciting place, but I think he exaggerated on how much I contributed. It was very interesting to listen to Shockley, Brattain, Bardeen, J. B. Johnson, Ken McKay and other people, and I was learning a lot. But unfortunately a Nobel Prize came, and a promotion came, and what was left was the dregs. Nobody wanted what was left. Well, there was no use eating with them!

Over on the other side of the dining hall was a chemistry table. I had worked with one of the fellows, Dave McCall; furthermore he was courting our secretary at the time. I went over and said, "Do you mind if I join you?" They can't say no, so I started eating with them for a while. And I started asking, "What are the important problems of your field?" And after a week or so, "What important problems are you working on?" And after some more time I came in one day and said, "If what you are doing is not important, and if you don't think it is going to lead to something important, why are you at Bell Labs working on it?" I wasn't welcomed after that; I had to find somebody else to eat with! That was in the spring.

In the fall, Dave McCall stopped me in the hall and said, "Hamming, that remark of yours got underneath my skin. I thought about it all summer, i.e. what were the important problems in my field. I haven't changed my research," he says, "but I think it was well worthwhile." And I said, "Thank you Dave," and went on. I noticed a couple of months later he was made the head of the department. I noticed the other day he was a Member of the National Academy of Engineering. I noticed he has succeeded. I have never heard the names of any of the other fellows at that table mentioned in science and scientific circles. They were unable to ask themselves, "What are the important problems in my field?"

If you do not work on an important problem, it's unlikely you'll do important work. It's perfectly obvious. Great scientists have thought through, in a careful way, a number of important problems in their field, and they keep an eye on wondering how to attack them. Let me warn you, 'important problem' must be phrased carefully. The three outstanding problems in physics, in a certain sense, were never worked on while I was at Bell Labs. By important I mean guaranteed a Nobel Prize and any sum of money you want to mention. We didn't work on (1) time travel, (2) teleportation, and (3) antigravity. They are not important problems because we do not have an attack. It's not the consequence that makes a problem important, it is that you have a reasonable attack. That is what makes a problem important. When I say that most scientists don't work on important problems, I mean it in that sense. The average scientist, so far as I can make out, spends almost all his time working on problems which he believes will not be important and he also doesn't believe that they will lead to important problems.

I spoke earlier about planting acorns so that oaks will grow. You can't always know exactly where to be, but you can keep active in places where something might happen. And even if you believe that great science is a matter of luck, you can stand on a mountain top where lightning strikes; you don't have to hide in the valley where you're safe. But the average scientist does routine safe work almost all the time and so he (or she) doesn't produce much. It's that simple. If you want to do great work, you clearly must work on important problems, and you should have an idea.

Along those lines at some urging from John Tukey and others, I finally adopted what I called "Great Thoughts Time." When I went to lunch Friday noon, I would only discuss great thoughts after that. By great thoughts I mean ones like: "What will be the role of computers in all of AT&T?", "How will computers change science?" For example, I came up with the observation at that time that nine out of ten experiments were done in the lab and one in ten on the computer. I made a remark to the vice presidents one time, that it would be reversed, i.e. nine out of ten experiments would be done on the computer and one in ten in the lab. They knew I was a crazy mathematician and had no sense of reality. I knew they were wrong and they've been proved wrong while I have been proved right. They built laboratories when they didn't need them. I saw that computers were transforming science because I spent a lot of time asking "What will be the impact of computers on science and how can I change it?" I asked myself, "How is it going to change Bell Labs?" I remarked one time, in the same address, that more than one-half of the people at Bell Labs will be interacting closely with computing machines before I leave. Well, you all have terminals now. I thought hard about where was my field going, where were the opportunities, and what were the important things to do. Let me go there so there is a chance I can do important things.

Most great scientists know many important problems. They have something between 10 and 20 important problems for which they are looking for an attack. And when they see a new idea come up, one hears them say "Well that bears on this problem." They drop all the other things and get after it. Now I can tell you a horror story that was told to me but I can't vouch for the truth of it. I was sitting in an airport talking to a friend of mine from Los Alamos about how it was lucky that the fission experiment occurred over in Europe when it did because that got us working on the atomic bomb here in the US. He said "No; at Berkeley we had gathered a bunch of data; we didn't get around to reducing it because we were building some more equipment, but if we had reduced that data we would have found fission." They had it in their hands and they didn't pursue it. They came in second!

The great scientists, when an opportunity opens up, get after it and they pursue it. They drop all other things. They get rid of other things and they get after an idea because they had already thought the thing through. Their minds are prepared; they see the opportunity and they go after it. Now of course lots of times it doesn't work out, but you don't have to hit many of them to do some great science. It's kind of easy. One of the chief tricks is to live a long time!

Another trait, it took me a while to notice. I noticed the following facts about people who work with the door open or the door closed. I notice that if you have the door to your office closed, you get more work done today and tomorrow, and you are more productive than most. But 10 years later somehow you don't know quite know what problems are worth working on; all the hard work you do is sort of tangential in importance. He who works with the door open gets all kinds of interruptions, but he also occasionally gets clues as to what the world is and what might be important. Now I cannot prove the cause and effect sequence because you might say, "The closed door is symbolic of a closed mind." I don't know. But I can say there is a pretty good correlation between those who work with the doors open and those who ultimately do important things, although people who work with doors closed often work harder. Somehow they seem to work on slightly the wrong thing—not much, but enough that they miss fame.

I want to talk on another topic. It is based on the song which I think many of you know, "It ain't what you do, it's the way that you do it." I'll start with an example of my own. I was conned into doing on a digital computer, in the absolute binary days, a problem which the best analog computers couldn't do. And I was getting an answer. When I thought carefully and said to myself, "You know, Hamming, you're going to have to file a report on this military job; after you spend a lot of money you're going to have to account for it and every analog installation is going to want the report to see if they can't find flaws in it." I was doing the required integration by a rather crummy method, to say the least, but I was getting the answer. And I realized that in truth the problem was not just to get the answer; it was to demonstrate for the first time, and beyond question, that I could beat the analog computer on its own ground with a digital machine. I reworked the method of solution, created a theory which was nice and elegant, and changed the way we computed the answer; the results were no different. The published report had an elegant method which was later known for years as "Hamming's Method of Integrating Differential Equations." It is somewhat obsolete now, but for a while it was a very good method. By changing the problem slightly, I did important work rather than trivial work.

In the same way, when using the machine up in the attic in the early days, I was solving one problem after another after another; a fair number were successful and there were a few failures. I went home one Friday after finishing a problem, and curiously enough I wasn't happy; I was depressed. I could see life being a long sequence of one problem after another after another. After quite a while of thinking I decided, "No, I should be in the mass production of a variable product. I should be concerned with *all* of next year's problems, not just the one in front of my face." By changing the question I still got the same kind of results or better, but I changed things and did important work. I attacked the major problem—How do I conquer machines and do all of next year's problems when I don't know what they are going to be? How do I prepare for it? How do I do this one so I'll be on top of it? How do

I obey Newton's rule? He said, "If I have seen further than others, it is because I've stood on the shoulders of giants." These days we stand on each other's feet!

You should do your job in such a fashion that others can build on top of it, so they will indeed say, "Yes, I've stood on so and so's shoulders and I saw further." The essence of science is cumulative. By changing a problem slightly you can often do great work rather than merely good work. Instead of attacking isolated problems, I made the resolution that I would never again solve an isolated problem except as characteristic of a class.

Now if you are much of a mathematician you know that the effort to generalize often means that the solution is simple. Often by stopping and saying, "This is the problem he wants but this is characteristic of so and so. Yes, I can attack the whole class with a far superior method than the particular one because I was earlier embedded in needless detail." The business of abstraction frequently makes things simple. Furthermore, I filed away the methods and prepared for the future problems.

To end this part, I'll remind you, "It is a poor workman who blames his tools–the good man gets on with the job, given what he's got, and gets the best answer he can." And I suggest that by altering the problem, by looking at the thing differently, you can make a great deal of difference in your final productivity because you can either do it in such a fashion that people can indeed build on what you've done, or you can do it in such a fashion that the next person has to essentially duplicate again what you've done. It isn't just a matter of the job, it's the way you write the report, the way you write the paper, the whole attitude. It's just as easy to do a broad, general job as one very special case. And it's much more satisfying and rewarding!

I have now come down to a topic which is very distasteful; it is not sufficient to do a job, you have to sell it. "Selling" to a scientist is an awkward thing to do. It's very ugly; you shouldn't have to do it. The world is supposed to be waiting, and when you do something great, they should rush out and welcome it. But the fact is everyone is busy with their own work. You must present it so well that they will set aside what they are doing, look at what you've done, read it, and come back and say, "Yes, that was good." I suggest that when you open a journal, as you turn the pages, you ask why you read some articles and not others. You had better write your report so when it is published in the Physical Review, or wherever else you want it, as the readers are turning the pages they won't just turn your pages but they will stop and read yours. If they don't stop and read it, you won't get credit.

There are three things you have to do in selling. You have to learn to write clearly and well so that people will read it, you must learn to give reasonably formal talks, and you also must learn to give informal talks. We had a lot of so-called "back room scientists." In a conference, they would keep quiet. Three weeks later after a decision was made they filed a report saying why you should do so and so. Well, it was too late. They would not stand up right in the middle of a hot conference, in the middle of activity, and say, "We should do this for these reasons." You need to master that form of communication as well as prepared speeches.

When I first started, I got practically physically ill while giving a speech, and I was very, very nervous. I realized I either had to learn to give speeches smoothly or I would essentially partially cripple my whole career. The first time IBM asked me to give a speech in New York one evening, I decided I was going to give a really good speech, a speech that was wanted, not a technical one but a broad one, and at the end if they liked it, I'd quietly say, "Any time you want one I'll come in and give you one." As a result, I got a great deal of practice giving speeches to a limited audience and I got over being afraid. Furthermore, I could also then study what methods were effective and what were ineffective.

While going to meetings I had already been studying why some papers are remembered and most are not. The technical person wants to give a highly limited technical talk. Most of the time the audience wants a broad general talk and wants much more survey and background than the speaker is willing to give. As a result, many talks are ineffective. The speaker names a topic and suddenly plunges into the details he's solved. Few people in the audience may follow. You should paint a general picture to say why it's important, and then slowly give a sketch of what was done. Then a larger number of people will say, "Yes, Joe has done that," or "Mary has done that; I really see where it is; yes, Mary really gave a good talk; I understand what Mary has done." The tendency is to give a highly restricted, safe talk; this is usually ineffective. Furthermore, many talks are filled with far too much information. So I say this idea of selling is obvious.

Let me summarize. You've got to work on important problems. I deny that it is all luck, but I admit there is a fair element of luck. I subscribe to Pasteur's "Luck favors the prepared mind." I favor heavily what I did. Friday afternoons for years—great thoughts only—means that I committed 10% of my time trying to understand the bigger problems in the field, i.e. what was and what was not important. I found in the early days I had believed "this" and yet had spent all week marching in "that" direction. It was kind of foolish. If I really believe the action is over there, why do I march in this direction? I either had to change my goal or change what I did. So I changed something I did and I marched in the direction I thought was important. It's that easy.

Now you might tell me you haven't got control over what you have to work on. Well, when you first begin, you may not. But once you're moderately successful, there are more people asking for results than you can deliver and you have some power of choice, but not completely. I'll tell you a story about that, and it bears on the subject of educating your boss. I had a boss named Schelkunoff; he was, and still is, a very good friend of mine. Some military person came to me and demanded some answers by Friday. Well, I had already dedicated my computing resources to reducing data on the fly for a group of scientists; I was knee deep in short, small, important problems. This military person wanted me to solve his problem by the end of the day on Friday. I said, "No, I'll give it to you Monday. I can work on it over the weekend. I'm not going to do it now." He goes down to my boss, Schelkunoff, and Schelkunoff says, "You must run this for him; he's got to have it by Friday." I tell him, "Why do I?"; he says, "You have to." I said, "Fine, Sergei, but you're sitting in your office Friday afternoon catching the late bus home to watch as this fellow walks out that door." I gave the military person the answers late Friday afternoon. I then went to Schelkunoff's office and sat down; as the man goes out I say, "You see Schelkunoff, this fellow has nothing under his arm; but I gave him the answers." On

Monday morning Schelkunoff called him up and said, "Did you come in to work over the weekend?" I could hear, as it were, a pause as the fellow ran through his mind of what was going to happen; but he knew he would have had to sign in, and he'd better not say he had when he hadn't, so he said he hadn't. Ever after that Schelkunoff said, "You set your deadlines; you can change them."

One lesson was sufficient to educate my boss as to why I didn't want to do big jobs that displaced exploratory research and why I was justified in not doing crash jobs which absorb all the research computing facilities. I wanted instead to use the facilities to compute a large number of small problems. Again, in the early days, I was limited in computing capacity and it was clear, in my area, that a "mathematician had no use for machines." But I needed more machine capacity. Every time I had to tell some scientist in some other area, "No I can't; I haven't the machine capacity," he complained. I said "Go tell *your* Vice President that Hamming needs more computing capacity." After a while I could see what was happening up there at the top; many people said to my Vice President, "Your man needs more computing capacity." I got it!

I also did a second thing. When I loaned what little programming power we had to help in the early days of computing, I said, "We are not getting the recognition for our programmers that they deserve. When you publish a paper you will thank that programmer or you aren't getting any more help from me. That programmer is going to be thanked by name; she's worked hard." I waited a couple of years. I then went through a year of BSTJ articles and counted what fraction thanked some programmer. I took it into the boss and said, "That's the central role computing is playing in Bell Labs; if the BSTJ is important, that's how important computing is." He had to give in. You can educate your bosses. It's a hard job. In this talk I'm only viewing from the bottom up; I'm not viewing from the top down. But I am telling you how you can get what you want in spite of top management. You have to sell your ideas there also.

Well I now come down to the topic, "Is the effort to be a great scientist worth it?" To answer this, you must ask people. When you get beyond their modesty, most people will say, "Yes, doing really first-class work, and knowing it, is as good as wine, women and song put together," or if it's a woman she says, "It is as good as wine, men and song put together." And if you look at the bosses, they tend to come back or ask for reports, trying to participate in those moments of discovery. They're always in the way. So evidently those who have done it, want to do it again. But it is a limited survey. I have never dared to go out and ask those who didn't do great work how they felt about the matter. It's a biased sample, but I still think it is worth the struggle. I think it is very definitely worth the struggle to try and do first-class work because the truth is, the value is in the struggle more than it is in the result. The struggle to make something of yourself seems to be worthwhile in itself. The success and fame are sort of dividends, in my opinion.

I've told you how to do it. It is so easy, so why do so many people, with all their talents, fail? For example, my opinion, to this day, is that there are in the mathematics department at Bell Labs quite a few people far more able and far better endowed than I, but they didn't produce as much. Some of them did produce more than I did;

Shannon produced more than I did, and some others produced a lot, but I was highly productive against a lot of other fellows who were better equipped. Why is it so? What happened to them? Why do so many of the people who have great promise, fail?

Well, one of the reasons is drive and commitment. The people who do great work with less ability but who are committed to it, get more done that those who have great skill and dabble in it, who work during the day and go home and do other things and come back and work the next day. They don't have the deep commitment that is apparently necessary for really first-class work. They turn out lots of good work, but we were talking, remember, about first-class work. There is a difference. Good people, very talented people, almost always turn out good work. We're talking about the outstanding work, the type of work that gets the Nobel Prize and gets recognition.

The second thing is, I think, the problem of personality defects. Now I'll cite a fellow whom I met out in Irvine. He had been the head of a computing center and he was temporarily on assignment as a special assistant to the president of the university. It was obvious he had a job with a great future. He took me into his office one time and showed me his method of getting letters done and how he took care of his correspondence. He pointed out how inefficient the secretary was. He kept all his letters stacked around there; he knew where everything was. And he would, on his word processor, get the letter out. He was bragging how marvelous it was and how he could get so much more work done without the secretary's interference. Well, behind his back, I talked to the secretary. The secretary said, "Of course I can't help him; I don't get his mail. He won't give me the stuff to log in; I don't know where he puts it on the floor. Of course I can't help him." So I went to him and said, "Look, if you adopt the present method and do what you can do single-handedly, you can go just that far and no farther than you can do single-handedly. If you will learn to work with the system, you can go as far as the system will support you." And, he never went any further. He had his personality defect of wanting total control and was not willing to recognize that you need the support of the system.

You find this happening again and again; good scientists will fight the system rather than learn to work with the system and take advantage of all the system has to offer. It has a lot, if you learn how to use it. It takes patience, but you can learn how to use the system pretty well, and you can learn how to get around it. After all, if you want a decision "No", you just go to your boss and get a "No" easy. If you want to do something, don't ask, do it. Present him with an accomplished fact. Don't give him a chance to tell you "No". But if you want a "No", it's easy to get a "No".

Another personality defect is ego assertion and I'll speak in this case of my own experience. I came from Los Alamos and in the early days I was using a machine in New York at 590 Madison Avenue where we merely rented time. I was still dressing in western clothes, big slash pockets, a bolo and all those things. I vaguely noticed that I was not getting as good service as other people. So I set out to measure. You came in and you waited for your turn; I felt I was not getting a fair deal. I said to myself, "Why? No Vice President at IBM said, "Give Hamming a bad time". It is the secretaries at the bottom who are doing this. When a slot appears, they'll rush to find someone to slip in, but they go out and find somebody else. Now, why? I haven't mistreated them." Answer, I wasn't dressing the way they felt somebody in that situation should. It came down to just that–I wasn't dressing properly. I had to make the decision–was I going to assert my ego and dress the way I wanted to and have it steadily drain my effort from my professional life, or was I going to appear to conform better? I decided I would make an effort to appear to conform properly. The moment I did, I got much better service. And now, as an old colorful character, I get better service than other people.

You should dress according to the expectations of the audience spoken to. If I am going to give an address at the MIT computer center, I dress with a bolo and an old corduroy jacket or something else. I know enough not to let my clothes, my appearance, my manners get in the way of what I care about. An enormous number of scientists feel they must assert their ego and do their thing their way. They have got to be able to do this, that, or the other thing, and they pay a steady price.

John Tukey almost always dressed very casually. He would go into an important office and it would take a long time before the other fellow realized that this is a first-class man and he had better listen. For a long time John has had to overcome this kind of hostility. It's wasted effort! I didn't say you should conform; I said "The *appearance of conforming* gets you a long way." If you chose to assert your ego in any number of ways, "I am going to do it my way," you pay a small steady price throughout the whole of your professional career. And this, over a whole lifetime, adds up to an enormous amount of needless trouble.

By taking the trouble to tell jokes to the secretaries and being a little friendly, I got superb secretarial help. For instance, one time for some idiot reason all the reproducing services at Murray Hill were tied up. Don't ask me how, but they were. I wanted something done. My secretary called up somebody at Holmdel, hopped the company car, made the hour-long trip down and got it reproduced, and then came back. It was a payoff for the times I had made an effort to cheer her up, tell her jokes and be friendly; it was that little extra work that later paid off for me. By realizing you have to use the system and studying how to get the system to do your work, you learn how to adapt the system to your desires. Or you can fight it steadily, as a small undeclared war, for the whole of your life.

And I think John Tukey paid a terrible price needlessly. He was a genius anyhow, but I think it would have been far better, and far simpler, had he been willing to conform a little bit instead of ego asserting. He is going to dress the way he wants all of the time. It applies not only to dress but to a thousand other things; people will continue to fight the system. Not that you shouldn't occasionally!

When they moved the library from the middle of Murray Hill to the far end, a friend of mine put in a request for a bicycle. Well, the organization was not dumb. They waited awhile and sent back a map of the grounds saying, "Will you please indicate on this map what paths you are going to take so we can get an insurance policy covering you." A few more weeks went by. They then asked, "Where are you going to store the bicycle and how will it be locked so we can do so and so." He finally realized that of course he was going to be red-taped to death so he gave in. He rose to be the President of Bell Laboratories.

Barney Oliver was a good man. He wrote a letter one time to the IEEE. At that time the official shelf space at Bell Labs was so much and the height of the IEEE Proceedings at that time was larger; and since you couldn't change the size of the official shelf space he wrote this letter to the IEEE Publication person saying, "Since so many IEEE members were at Bell Labs and since the official space was so high the journal size should be changed." He sent it for his boss's signature. Back came a carbon with his signature, but he still doesn't know whether the original was sent or not. I am not saying you shouldn't make gestures of reform. I am saying that my study of able people is that they don't get themselves *committed* to that kind of warfare. They play it a little bit and drop it and get on with their work.

Many a second-rate fellow gets caught up in some little twitting of the system, and carries it through to warfare. He expends his energy in a foolish project. Now you are going to tell me that somebody has to change the system. I agree; somebody's has to. Which do you want to be? The person who changes the system or the person who does first-class science? Which person is it that you want to be? Be clear, when you fight the system and struggle with it, what you are doing, how far to go out of amusement, and how much to waste your effort fighting the system. My advice is to let somebody else do it and you get on with becoming a first-class scientist. Very few of you have the ability to both reform the systemand become a first-class scientist.

On the other hand, we can't always give in. There are times when a certain amount of rebellion is sensible. I have observed almost all scientists enjoy a certain amount of twitting the system for the sheer love of it. What it comes down to basically is that you cannot be original in one area without having originality in others. Originality is being different. You can't be an original scientist without having some other original characteristics. But many a scientist has let his quirks in other places make him pay a far higher price than is necessary for the ego satisfaction he or she gets. I'm not against all ego assertion; I'm against some.

Another fault is anger. Often a scientist becomes angry, and this is no way to handle things. Amusement, yes, anger, no. Anger is misdirected. You should follow and cooperate rather than struggle against the system all the time.

Another thing you should look for is the positive side of things instead of the negative. I have already given you several examples, and there are many, many more; how, given the situation, by changing the way I looked at it, I converted what was apparently a defect to an asset. I'll give you another example. I am an egotistical person; there is no doubt about it. I knew that most people who took a sabbatical to write a book, didn't finish it on time. So before I left, I told all my friends that when I come back, that book was going to be done! Yes, I would have it done–I'd have been ashamed to come back without it! I used my ego to make myself behave the way I wanted to. I bragged about something so I'd have to perform. I found out many times, like a cornered rat in a real trap, I was surprisingly capable. I have found that it paid to say, "Oh yes, I'll get the answer for you Tuesday," not having any idea how to do it. By Sunday night I was really hard thinking on how I was going to deliver by Tuesday. I often put my pride on the line and sometimes I failed, but as I said, like a cornered rat I'm surprised how often I did a good job. I think you need to learn to use yourself. I think you need to know how to convert a situation from one view to another which would increase the chance of success.

Now self-delusion in humans is very, very common. There are enumerable ways of you changing a thing and kidding yourself and making it look some other way. When you ask, "Why didn't you do such and such," the person has a thousand alibis. If you look at the history of science, usually these days there are 10 people right there ready, and we pay off for the person who is there first. The other nine fellows say, "Well, I had the idea but I didn't do it and so on and so on." There are so many alibis. Why weren't you first? Why didn't you do it right? Don't try an alibi. Don't try and kid yourself. You can tell other people all the alibis you want. I don't mind. But to yourself try to be honest.

If you really want to be a first-class scientist you need to know yourself, your weaknesses, your strengths, and your bad faults, like my egotism. How can you convert a fault to an asset? How can you convert a situation where you haven't got enough manpower to move into a direction when that's exactly what you need to do? I say again that I have seen, as I studied the history, the successful scientist changed the viewpoint and what was a defect became an asset.

In summary, I claim that some of the reasons why so many people who have greatness within their grasp don't succeed are: they don't work on important problems, they don't become emotionally involved, they don't try and change what is difficult to some other situation which is easily done but is still important, and they keep giving themselves alibis why they don't. They keep saying that it is a matter of luck. I've told you how easy it is; furthermore I've told you how to reform. Therefore, go forth and become great scientists!

(End of the formal part of the talk.)

Discussion–Questions and answers

A. G. Chynoweth: Well that was 50 minutes of concentrated wisdom and observations accumulated over a fantastic career; I lost track of all the observations that were striking home. Some of them are very very timely. One was the plea for more computer capacity; I was hearing nothing but that this morning from several people, over and over again. So that was right on the mark today even though here we are 20 – 30 years after when you were making similar remarks, Dick. I can think of all sorts of lessons that all of us can draw from your talk. And for one, as I walk around the halls in the future I hope I won't see as many closed doors in Bellcore. That was one observation I thought was very intriguing. Thank you very, very much indeed Dick; that was a wonderful recollection. I'll now open it up for questions. I'm sure there are many people who would like to take up on some of the points that Dick was making.

Hamming: First let me respond to Alan Chynoweth about computing. I had computing in research and for 10 years I kept telling my management, "Get that !&@#% machine out of research. We are being forced to run problems all the time. We can't do research because were too busy operating and running the computing machines." Finally the message got through. They were going to move computing out of research to someplace else. I was persona non grata to say the least and I was surprised that people didn't kick my shins because everybody was having their toy taken away from them. I went in to Ed David's office and said, "Look Ed, you've got to give your researchers a machine. If you give them a great big machine, we'll be back in the same trouble we were before, so busy keeping it going we can't think. Give them the smallest machine you can because they are very able people. They will learn how to do things on a small machine instead of mass computing." As far as I'm concerned, that's how UNIX arose. We gave them a moderately small machine and they decided to make it do great things. They had to come up with a system to do it on. It is called UNIX!

A. G. Chynoweth: I just have to pick up on that one. In our present environment, Dick, while we wrestle with some of the red tape attributed to, or required by, the regulators, there is one quote that one exasperated AVP came up with and I've used it over and over again. He growled that, "UNIX was never a deliverable!"

Question: What about personal stress? Does that seem to make a difference?

Hamming: Yes, it does. If you don't get emotionally involved, it doesn't. I had incipient ulcers most of the years that I was at Bell Labs. I have since gone off to the Naval Postgraduate School and laid back somewhat, and now my health is much better. But if you want to be a great scientist you're going to have to put up with stress. You can lead a nice life; you can be a nice guy or you can be a great scientist. But nice guys end last, is what Leo Durocher said. If you want to lead a nice happy life with a lot of recreation and everything else, you'll lead a nice life.

Question: The remarks about having courage, no one could argue with; but those of us who have gray hairs or who are well established don't have to worry too much. But what I sense among the young people these days is a real concern over the risk taking in a highly competitive environment. Do you have any words of wisdom on this?

Hamming: I'll quote Ed David more. Ed David was concerned about the general loss of nerve in our society. It does seem to me that we've gone through various periods. Coming out of the war, coming out of Los Alamos where we built the bomb, coming out of building the radars and so on, there came into the mathematics department, and the research area, a group of people with a lot of guts. They've just seen things done; they've just won a war which was fantastic. We had reasons for having courage and therefore we did a great deal. I can't arrange that situation to do it again. I cannot blame the present generation for not having it, but I agree with what you say; I just cannot attach blame to it. It doesn't seem to me they have the desire for greatness; they lack the courage to do it. But we had, because we were in a favorable circumstance to have it; we just came through a tremendously successful war. In the war we were looking very, very bad for a long while; it was a very desperate struggle as you well know. And our success, I think, gave us courage and self confidence; that's why you see, beginning in the late forties through the fifties, a tremendous productivity at the labs which was stimulated from the earlier times. Because many of us were earlier forced to learn other things–we were forced to learn the things we didn't want to learn, we were forced to have an open door–and then we could exploit those things we learned. It is true, and I can't do anything about it; I cannot blame the present generation either. It's just a fact.

Question: Is there something management could or should do?

Hamming: Management can do very little. If you want to talk about managing research, that's a totally different talk. I'd take another hour doing that. This talk is about how the individual gets very successful research done in spite of anything the management does or in spite of any other opposition. And how do you do it? Just as I observe people doing it. It's just that simple and that hard!

Question: Is brainstorming a daily process?

Hamming: Once that was a very popular thing, but it seems not to have paid off. For myself I find it desirable to talk to other people; but a session of brainstorming is seldom worthwhile. I do go in to strictly talk to somebody and say, "Look, I think there has to be something here. Here's what I think I see. . . " and then begin talking back and forth. But you want to pick capable people. To use another analogy, you know the idea called the "critical mass." If you have enough stuff you have critical mass. There is also the idea I used to call "sound absorbers". When you get too many sound absorbers, you give out an idea and they merely say, "Yes, yes, yes." What you want to do is get that critical mass in action; "Yes, that reminds me of so and so," or, "Have you thought about that or this?" When you talk to other people, you want to get rid of those sound absorbers who are nice people but merely say, "Oh yes," and to find those who will stimulate you right back.

For example, you couldn't talk to John Pierce without being stimulated very quickly. There were a group of other people I used to talk with. For example there was Ed Gilbert; I used to go down to his office regularly and ask him questions and listen and come back stimulated. I picked my people carefully with whom I did or whom I didn't brainstorm because the sound absorbers are a curse. They are just nice guys; they fill the whole space and they contribute nothing except they absorb ideas and the new ideas just die away instead of echoing on. Yes, I find it necessary to talk to people. I think people with closed doors fail to do this so they fail to get their ideas sharpened, such as "Did you ever notice something over here?" I never knew anything about it–I can go over and look. Somebody points the way. On my visit here, I have already found several books that I must read when I get home. I talk to people and ask questions when I think they can answer me and give me clues that I do not know about. I go out and look!

Question: What kind of tradeoffs did you make in allocating your time for reading and writing and actually doing research?

Hamming: I believed, in my early days, that you should spend at least as much time in the polish and presentation as you did in the original research. Now at least 50% of the time must go for the presentation. It's a big, big number.

Question: How much effort should go into library work?

Hamming: It depends upon the field. I will say this about it. There was a fellow at Bell Labs, a very, very, smart guy. He was always in the library; he read everything. If you wanted references, you went to him and he gave you all kinds of references. But in the middle of forming these theories, I formed a proposition: there would be no effect named after him in the long run. He is now retired from Bell Labs and is an

Adjunct Professor. He was very valuable; I'm not questioning that. He wrote some very good Physical Review articles; but there's no effect named after him because he read too much. If you read all the time what other people have done you will think the way they thought. If you want to think new thoughts that are different, then do what a lot of creative people do—get the problem reasonably clear and then refuse to look at any answers until you've thought the problem through carefully how you would do it, how you could slightly change the problem to be the correct one. So yes, you need to keep up. You need to keep up more to find out what the problems are than to read to find the solutions. The reading is necessary to know what is going on and what is possible. But reading to get the solutions does not seem to be the way to do great research. So I'll give you two answers. You read; but it is not the amount, it is the way you read that counts.

Question: How do you get your name attached to things?

Hamming: By doing great work. I'll tell you the hamming window one. I had given Tukey a hard time, quite a few times, and I got a phone call from him from Princeton to me at Murray Hill. I knew that he was writing up power spectra and he asked me if I would mind if he called a certain window a "Hamming window." And I said to him, "Come on, John; you know perfectly well I did only a small part of the work but you also did a lot." He said, "Yes, Hamming, but you contributed a lot of small things; you're entitled to some credit." So he called it the hamming window. Now, let me go on. I had twitted John frequently about true greatness. I said true greatness is when your name is like ampere, watt, and fourier—when it's spelled with a lower case letter. That's how the hamming window came about.

Question: Dick, would you care to comment on the relative effectiveness between giving talks, writing papers, and writing books?

Hamming: In the short-haul, papers are very important if you want to stimulate someone tomorrow. If you want to get recognition long-haul, it seems to me writing books is more contribution because most of us need orientation. In this day of practically infinite knowledge, we need orientation to find our way. Let me tell you what infinite knowledge is. Since from the time of Newton to now, we have come close to doubling knowledge every 17 years, more or less. And we cope with that, essentially, by specialization. In the next 340 years at that rate, there will be 20 doublings, i.e. a million, and there will be a million fields of specialty for every one field now. It isn't going to happen. The present growth of knowledge will choke itself off until we get different tools. I believe that books which try to digest, coordinate, get rid of the duplication, get rid of the less fruitful methods and present the underlying ideas clearly of what we know now, will be the things the future generations will value. Public talks are necessary; private talks are necessary; written papers are necessary. But I am inclined to believe that, in the long-haul, books which leave out what's not essential are more important than books which tell you everything because you don't want to know everything. I don't want to know that much about penguins is the usual reply. You just want to know the essence.

Question: You mentioned the problem of the Nobel Prize and the subsequent notoriety of what was done to some of the careers. Isn't that kind of a much more broad problem of fame? What can one do?

Hamming: Some things you could do are the following. Somewhere around every seven years make a significant, if not complete, shift in your field. Thus, I shifted from numerical analysis, to hardware, to software, and so on, periodically, because you tend to use up your ideas. When you go to a new field, you have to start over as a baby. You are no longer the big mukity muk and you can start back there and you can start planting those acorns which will become the giant oaks. Shannon, I believe, ruined himself. In fact when he left Bell Labs, I said, "That's the end of Shannon's scientific career." I received a lot of flak from my friends who said that Shannon was just as smart as ever. I said, "Yes, he'll be just as smart, but that's the end of his scientific career," and I truly believe it was.

You have to change. You get tired after a while; you use up your originality in one field. You need to get something nearby. I'm not saying that you shift from music to theoretical physics to English literature; I mean within your field you should shift areas so that you don't go stale. You couldn't get away with forcing a change every seven years, but if you could, I would require a condition for doing research, being that you will change your field of research every seven years with a reasonable definition of what it means, or at the end of 10 years, management has the right to compel you to change. I would insist on a change because I'm serious. What happens to the old fellows is that they get a technique going; they keep on using it. They were marching in that direction which was right then, but the world changes. There's the new direction; but the old fellows are still marching in their former direction.

You need to get into a new field to get new viewpoints, and *before* you use up all the old ones. You can do something about this, but it takes effort and energy. It takes courage to say, "Yes, I will give up my great reputation." For example, when error correcting codes were well launched, having these theories, I said, "Hamming, you are going to quit reading papers in the field; you are going to ignore it completely; you are going to try and do something else other than coast on that." I deliberately refused to go on in that field. I wouldn't even read papers to try to force myself to have a chance to do something else. I managed myself, which is what I'm preaching in this whole talk. Knowing many of my own faults, I manage myself. I have a lot of faults, so I've got a lot of problems, i.e. a lot of possibilities of management.

Question: Would you compare research and management?

Hamming: If you want to be a great researcher, you won't make it being president of the company. If you want to be president of the company, that's another thing. I'm not against being president of the company. I just don't want to be. I think Ian Ross does a good job as President of Bell Labs. I'm not against it; but you have to be clear on what you want. Furthermore, when you're young, you may have picked wanting to be a great scientist, but as you live longer, you may change your mind. For instance, I went to my boss, Bode, one day and said, "Why did you ever become department head? Why didn't you just be a good scientist?" He said, "Hamming, I had a vision of what mathematics should be in Bell Laboratories. And I saw if that vision was going to be realized, I had to make it happen; I had to be department head." When your vision of what you want to do is what you can do single-handedly, then you should pursue it. The day your vision, what you think needs to be done, is bigger than what you can do single-handedly, then you have to move toward management. And the bigger the vision is, the farther in management you have to go. If you have a vision of what the whole laboratory should be, or the whole Bell System, you have to get there to make it happen. You can't make it happen from the bottom very easily. It depends upon what goals and what desires you have. And as they change in life, you have to be prepared to change. I chose to avoid management because I preferred to do what I could do single-handedly. But that's the choice that I made, and it is biased. Each person is entitled to their choice. Keep an open mind. But when you do choose a path, for heaven's sake be aware of what you have done and the choice you have made. Don't try to do both sides.

Question: How important is one's own expectation or how important is it to be in a group or surrounded by people who expect great work from you?

Hamming: At Bell Labs everyone expected good work from me–it was a big help. Everybody expects you to do a good job, so you do, if you've got pride. I think it's very valuable to have first-class people around. I sought out the best people. The moment that physics table lost the best people, I left. The moment I saw that the same was true of the chemistry table, I left. I tried to go with people who had great ability so I could learn from them and who would expect great results out of me. By deliberately managing myself, I think I did much better than laissez faire.

Question: You, at the outset of your talk, minimized or played down luck; but you seemed also to gloss over the circumstances that got you to Los Alamos, that got you to Chicago, that got you to Bell Laboratories.

Hamming: There was some luck. On the other hand I don't know the alternate branches. Until you can say that the other branches would not have been equally or more successful, I can't say. Is it luck the particular thing you do? For example, when I met Feynman at Los Alamos, I knew he was going to get a Nobel Prize. I didn't know what for. But I knew darn well he was going to do great work. No matter what directions came up in the future, this man would do great work. And sure enough, he did do great work. It isn't that you only do a little great work at this circumstance and that was luck, there are many opportunities sooner or later. There are a whole pail full of opportunities, of which, if you're in this situation, you seize one and you're great over there instead of over here. There is an element of luck, yes and no. Luck favors a prepared mind; luck favors a prepared person. It is not guaranteed; I don't guarantee success as being absolutely certain. I'd say luck changes the odds, but there is some definite control on the part of the individual.

Go forth, then, and do great work!

(End of the General Research Colloquium Talk.)

Biographical sketch of Richard Hamming

Richard W. Hamming was born February 11, 1915, in Chicago, Illinois. His formal education was marked by the following degrees (all in mathematics): B.S. 1937, University of Chicago; M.A. 1939, University of Nebraska; and PhD 1942, University of Illinois. His early experience was obtained at Los Alamos 19451946, i.e. at the close of World War II, where he managed the computers used in building the first atomic bomb. From there he went directly to Bell Laboratories where he spent thirty years in various aspects of computing, numerical analysis, and management of computing, i.e. 1946–1976. On July 23, 1976 he "moved his office" to the Naval Postgraduate School in Monterey, California where he taught, supervised research, and wrote books.

While at Bell Laboratories, he took time to teach in Universities, sometimes locally and sometimes on a full sabbatical leave; these activities included visiting professorships at New York University, Princeton University (Statistics), City College of New York, Stanford University, 1960–61, Stevens Institute of Technology (Mathematics), and the University of California, Irvine, 1970–71.

Richard Hamming has received a number of awards which include: Fellow, IEEE, 1968; the ACM Turing Prize, 1968; the IEEE Emanuel R. Piore Award, 1979; Member, National Academy of Engineering, 1980; and the Harold Pender Award, U. Penn., 1981. In 1987 a major IEEE award was named after him, namely the Richard W. Hamming Medal, "For exceptional contributions to information sciences and systems"; fittingly, he was also the first recipient of this award, 1988. In 1996 in Munich he received the prestigious $130,000 Eduard Rhein Award for Achievement in Technology for his work on error correcting codes. He was both a Founder and Past President of ACM, and a Vice Pres. of the AAAS Mathematics Section.

He is probably best known for his pioneering work on error-correcting codes, his work on integrating differential equations, and the spectral window which bears his name. His extensive writing has included a number of important, pioneering, and highly regarded books. These are:

- *Numerical Methods for Scientists and Engineers*,McGraw-Hill, 1962; Second edition 1973; Reprinted by Dover 1985; Translated into Russian.
- *Calculus and the Computer Revolution*, Houghton-Mifflin, 1968.
- *Introduction to Applied Numerical Analysis*, McGraw-Hill, 1971.
- *Computers and Society*, McGraw-Hill, 1972.
- *Digital Filters*, Prentice-Hall, 1977; Second edition 1983; Third edition 1989; translated into several European languages.
- *Coding and Information Theory*, Prentice-Hall, 1980; Second edition 1986.
- *Methods of Mathematics Applied to Calculus, Probability and Statistics*, Prentice-Hall, 1985.
- *The Art of Probability for Scientists and Engineers*, Addison-Wesley, 1991.
- *The Art of Doing Science and Engineering: Learning to Learn*, Gordon and Breach, 1997.

He continued a very active life as Adjunct Professor, teaching and writing in the Mathematics and Computer Science Departments at the Naval Postgraduate School, Monterey, California for another twenty- one years before he retired to become Professor Emeritus in 1997. He was still teaching a course in the fall of 1997. He passed away unexpectedly on January 7, 1998.

Acknowledgement

I would like to acknowledge the professional efforts of Donna Paradise of the Word Processing Center who did the initial transcription of the talk from the tape recording. She made my job of editing much easier. The errors of sentence parsing and punctuation are mine and mine alone. Finally I would like to express my sincere appreciation to Richard Hamming and Alan Chynoweth for all of their help in bringing this transcription to its present readable state.

J. F. Kaiser

James F. Kaiser received the E.E. degree from the U. of Cincinnati in 1952 and the S.M. and Sc.D. degrees from M.I.T. in 1954 and 1959 respectively, all in electrical engineering. He is currently a Visiting Professor in the Department of Electrical and Computer Engineering at Duke U., Durham, NC. He was formerly a Distinguished Member of Technical Staff with Bell Communications Research from 1984–1991. Prior to that he was a Distinguished Member of Technical Staff in the Research Division of Bell Laboratories, Murray Hill, NJ from 1959–1984 where he did research in the areas of digital signal processing, system simulation, computer graphics, and computer-aided design. He is the author of more than 65 research papers and the coauthor and editor of eight books in the signal processing and automatic control areas.

Dr. Kaiser is a Life Fellow of the IEEE, a Fellow of the AAAS, and a member of SIAM and the Acoustical Society of America. In addition he has received the Society Award, the Technical Achievement Award, and the Meritorious Service Award all from the IEEE Signal Processing Society. From the University of Cincinnati, College of Engineering he received their Distinguished Engineering Alumnus Award and their Eta Kappa Nu Award of Merit. In 2000 he was honored with the IEEE Jack S. Kilby Medal for his pioneering work in digital signal processing.

7

SIMULA RESEARCH LABORATORY — A DIFFERENT RESEARCH INSTITUTION

Marianne M. Sundet and Bjarne Røsjø

The idea for the research centre that was later to become Simula Research Laboratory was conceived in 1999, at the end of a long political debate about the future use of the site of the former Oslo airport. Since its establishment Simula has surprised both friends and critics with its excellent research results and its innovative approach to challenges and opportunities. This chapter provides a description of the background and main features of Simula Research Laboratory.

A unique research organisation

A novelty when it was established, Simula Research Laboratory is an institution that stands out in the Norwegian research system. The original vision was to create a research centre characterised by excellence. To this end, Simula has been provided with a long-term financing model, unique compared to the Norwegian tradition. As a result, researchers at Simula are not required to spend their time constantly pursuing funding, as is the case at the traditional research institutes. Nor do they have to take on onerous teaching and supervising duties as at the Norwegian universities where

Marianne M. Sundet
Simula Research Laboratory

Bjarne Røsjø
Freelance science writer and communications consultant

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_7,

time is a limited commodity. The generous funding that kick-started and continues to fuel Simula allows a strong focus on scientific quality within its specific subject areas. On this basis, Simula aims to be an efficient producer of strong research results. To achieve this goal, one primary objective has been to bring back the full-time researcher, to allow excellent researchers to devote their time to research—and spend substantially less time worrying about funding, teaching and supervising.

Simula organises and manages its research and other activities in a different way from most other research organizations in Norway. It is owned by the Norwegian government, the independent research organisation SINTEF and Norwegian Computing Center. It is run like a private company, in accordance with the Limited Companies Act. In comparison with the universities, Simula places greater emphasis on the possible applications of its research. In addition, at Simula, the researchers work collectively towards a determined set of long-term research goals, addressing fundamental problems that are important for society and industry. Thus individual researchers at Simula have less freedom to choose research topics than their counterparts at the universities. However, having observed the substantial volume of strong results that have blossomed over just a few years, most people working in Simula see this payment as an investment. The results, as exemplified in this book, speak for themselves, and the differently flavoured research institution has been well received by important policy-makers. "There is a need for different kinds of research institutions; universities, university colleges and distinctive organisations like Simula. The way Simula combines research and education with collaboration with industry is impressive," said Minister of Research and Higher Education Tora Aasland when she visited Simula in June 2008.

The name Simula Research Laboratory originates from a major Norwegian scientific achievement: the development in the early 1960s of the programming language Simula at the Norwegian Computing Center. This work was carried out by Kristen Nygaard and Ole-Johan Dahl, who were later appointed professors at the Department of Informatics, University of Oslo. Nygaard and Dahl were awarded both the A.M. Turing Prize for 2001 and the John von Neumann Medal for 2002, for the development of the language Simula and the concept of object-oriented programming. The two awards are the nearest you come to a Nobel Prize in informatics, and are awarded by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE), respectively. The programming language Simula and the concept of object-oriented programming are explained further in chapter 11.

The idea – ICT research at the old airport

A long political debate started in the early 1990s after the Norwegian Parliament had decided that the international airport at Fornebu was to be closed down, and that a new airport should be built at Gardermoen, north of Oslo. The final decision was taken by the Parliament in March 2000, but that is another story that you can read more about in chapter 8 on page 77. Both politically and financially, the legislators supported the idea of creating an ICT and knowledge park at the site of the former

From the left: Director of Research Education Kristin Vinje, Minister of Research and Higher Education Tora Aasland, and Managing Director Aslak Tveito. In the background: The minister's political adviser Kyrre Lekve. Photo: Simula Research Laboratory.

airport. Moreover, the Norwegian government decided that a research unit[1] should form the scientific core of the park. The government proposed that a total of 45 million Norwegian kroner would be allocated in the year 2000 for the establishment of this new research centre at Fornebu, jointly financed by the Ministry of Education, Research and Church Affairs (30 million NOK), the Ministry of Trade and Industry (10 million NOK), and the Ministry of Transport and Communications (5 million NOK). The Research Council of Norway was given the responsibility for coordinating the planning and establishment of the research centre, in close cooperation with existing institutions for research and education.

The idea that a centre of basic research should be incorporated into the new park came from Jon Lilletun, Minister of Education, Research and Church Affairs from 1997 to 2000. Professor Morten Dæhlen, chair of the interim board, describes, in chapter 8 on the history of IT Fornebu on page 86, how in 1999 he received an unexpected message from the Ministry: if an ICT and knowledge centre were to be set up at Fornebu the Minister was determined that it should also include a research unit that would form its scientific core.

[1] Report No. 1 to the Storting (1999–2000).

Setting up a research centre

The Research Council of Norway's first task was to decide what type of ICT research centre should be established at Fornebu, and who would be given the job. An interim board was set up for this purpose. The interim board appointed a committee, which identified a number of possible research areas for the new research centre, based on existing ICT research activities at the Norwegian universities. The relevant Norwegian research communities were invited to submit proposals, and a competition between 12 applicant groups ensued.

The interim board also led the application assessment process, and delivered its proposal to the Research Council in November 2000. Three research groups were subsequently selected; two of the groups on the basis of the potential of their topic to help create an IT and knowledge centre at Fornebu, the third on the basis of scientific quality. The research centre was given the task of setting up research departments based on the three selected groups. With the establishment of the new centre, the three research groups, all from the Department of Informatics at the University of Oslo, were given opportunities to target their research in a new way. Even today, the departmental structure of Simula's basic research unit can be directly traced back to the selection of research.

The interim board proposed that the centre be set up as a limited company, owned jointly by the University of Oslo, the Norwegian University of Science and Technology, the University of Bergen, the University of Tromsø, SINTEF and the Norwegian Computing Center. The Research Council approved this proposal, and the owners were asked to nominate representatives to the Board. Later the Government decided that the State should be the owner of the centre, representing the Universities. A collaboration agreement outlining the way Simula was to be run was drawn up between the owners and the Research Council, and agreements regulating the relationship between Simula and each of the individual owners were entered into. A ten-year research agreement (2001–2010) was signed with the Government, represented by the Research Council, stating that the Simula centre was to be evaluated at the end of a five-year period. In addition, agreements with the Department of Informatics at UiO, concerning the teaching and supervision of graduate students, were also drawn up.

In 2001, before it had been set up as a limited company the Simula centre was organised as a project at UiO. At this time, the management concentrated on hiring personnel and planning the move from the centre's temporary location at Oslo Innovation Centre at Gaustadbekkdalen close to UiO's campus, to its new offices at Fornebu. At a meeting at the Ministry of Education and Research on 11 June 2002, Simula was established as a limited company, and only six days later, the new Board of Directors consisting of representatives of the owners, convened for the first time.

The first managing director of Simula was Professor Morten Dæhlen from the Department of Informatics at UiO. Professor Aslak Tveito has served as managing director since 2002[2].

[2] An overview of management positions is listed in Appedix A.

The three research departments

The main objective was that within three to five years the research groups at Simula should be performing basic research at a high international level. This objective necessitated publications in international journals, and visibility on the international conference scene, but most of all a strong focus on the selected research topics. Refining the scientific profile of the university groups that were selected to form Simula, a decision was taken to concentrate research efforts on communication technology, scientific computing and software engineering. These three designated subject areas still represent the focus of Simula's research and form the basis for the three research departments.

The Networks and Distributed Systems department conducts research on a wide range of communication networks, from networks on chips to Internet backbones and wireless networks. The department's main concern is to improve the performance and resilience of these networks. The Scientific Computing department conducts research on numerical methods and software for computer-based simulation of physical processes arising in select areas of biomedicine and geoscience. There is also an increasing amount of research carried out on using the developed techniques to gain new scientific insight within the application areas, such as improved understanding of the electrophysiology of the heart, how to locate damaged heart tissue from ECG recordings, and reliable estimation of how sand and mud accumulates to form new layers of rock. All the projects are strongly interrelated with respect to numerical methods and software. The Software Engineering department aims to improve software engineering technologies, as regards processes, methods, tools, and frameworks. The main topics addressed are effort estimation, software maintenance, and testing. These are all areas of great importance to the efficiency and quality of the software industry. The department also runs a project on the improvement of empirical research methods. The activities in the research departments are further described in part II–Basic Research.

The official opening

At the official opening ceremony in April 2002, Kristin Clemet, then Minister of Education and Research, personally announced the necessary extra funding that would compensate for the budget cuts that Simula had experienced under the previous Minister, Trond Giske. In her speech[3] Clemet made it clear that the official opening was an important occasion for the Norwegian research community and that the Norwegian Government would provide funding to Simula at the level originally set. "With the Simula centre we have obtained a research community that prioritises quality and gives skilled researchers the time and resources necessary to carry out research of the highest international calibre. We will invest extensively in Norwegian

[3] The Minister's speech (in Norwegian only) can be found at: www.regjeringen.no/nb/dokumentarkiv/Regjeringen-Bondevik-II/ufd/265588/265391/tale_ved_offisiell_apning_av_simula-senteret.html?id=265445

research, but we must also be bold enough to concentrate our efforts on specific target areas."

Clemet also noted that Simula was focusing on the five elements identified as critical for the future of Norwegian research: quality, internationalisation, strong technical leadership, knowledge transfer, and researcher mobility. In addition, Simula was committed to maintaining a close relationship with industry. The then Ministers of Justice, Agriculture, Petroleum and Energy, Trade and Industry, and International Development were also present at the opening.

From the left: The rector of University of Oslo Arild Underdal, the Norwegian Minister of Justice Odd Einar Dørum, Simula's Managing Director Morten Dæhlen, the Minister of Education and Research Kristin Clemet, and the Minister of International Development Hilde Frafjord Johnson. Photo: Simula Research Laboratory.

The Scientific Advisory Board

The group of researchers who came together to form Simula in 2001 counted several well-respected professors from the University of Oslo. They all had considerable experience of scientific work and of running research projects, but none had experience of setting up and running an institution like Simula. It was clear that this would be a formidable task, with challenges related both to purely administrative issues, and to strategic decisions concerning the scientific work of such a large unit. Experienced administrative staff members were employed to deal with the former, and a Scientific Advisory Board (SAB) was set up to deal with strategy. It was decided that the SAB would consist of six members, two for each research department, who would meet annually and present their reports and recommendations to the Board. The SAB held its first meeting in March 2002[4]. This meeting, and the subsequent report, made it clear that the SAB mandate was in need of some clarification. It was subsequently decided that the SAB, instead of addressing the Board directly, should report to the managing director, and should act in an advisory capacity for Simula's administration and the department managers. Furthermore it was decided that in this role there was no need for the SAB members to be independent. Instead, they were encouraged to interact closely with their respective research departments.

The SAB held its second meeting in January 2003. This proved to be far more fruitful than the first. In its new advisory role, which did not necessitate independence, the SAB members were able to become more deeply involved in discussions with the research departments, and thereby provide stronger and more detailed advice. In particular, the report of the SAB was instrumental in defining the role and ambitions of SE within the field of empirical software engineering, in further focusing and targeting of the research in SC, and in clarifying the strategy of ND. The SAB also played an important role in the evaluation of Simula in 2004. The SAB wrote detailed reports both about Simula as a whole and about each research department, expressing its view of the results and recommending further developments. The significance of these SAB reports is illustrated by the fact that the report of the Evaluation Committee supported several of the most important recommendations made by the SAB[5].

Taking steps to secure further funding

Originally, the ten-year research agreement (2001–2010) stated that the Simula centre should be evaluated by a panel of international experts appointed by the Research Council in 2004/2005. Funding for the five-year period 2006–2010 would depend on the recommendations and findings of this evaluation. The evaluation was performed by an international committee consisting of five professors from the USA, Great Britain, Sweden and France. The Committee's report verified that the results of the establishment and growth phases were good. The report described Simula's scientific quality, operational efficiency, and visibility in the media and society in very positive terms[6]. It confirmed that Simula was a research unit that was gaining international recognition, and the Committee recommended to the Research Council that Sim-

[4] The first SAB had only four members; Malcolm Atkinson (SE), David Hutchison (ND), Rolf Jeltsch (SC), and Dieter Rombach (SE). Later the SAB was extended to include two new members; Jose Duato (ND) and Michael Thune (SC).

[5] A new SAB was set up in 2005 consisting of three members representing the research departments, Barbara Kitchenham (SE), Andrew McCulloch (SC), Klara Nahrstedt (ND), and two members with new roles: Professor Torger Reve, President of the Norwegian School of Management (1997–2005), to advise on innovation activities, and Professor Lars Walløe, University of Oslo, to advise on overall strategic issues.

[6] The report is available at simula.no/about/SimulaEvaluationReport_des04.pdf

ula's contract be renewed and that it be granted more long-term perspective on the funding. The Evaluation Committee confirmed that Simula had come far in the short time since it was established, but also pointed out potential areas for improvement.

Simula's fundamental activities are funded through a basic allowance from the state. The allowance from the Government is channelled through the Research Council of Norway. According to the original contract with the Research Council, the funding was aimed at 31 million kroner in 2001, and then 45 million kroner from 2002. In addition to the basic funding, the agreement with the Research Council is based on the expectation that Simula seek funding externally, from the Research Council's projects, industry, the EU and other relevant sources.

In the light of the positive evaluation in 2004, the Research Council decided to extend its contract with Simula until 2010. The Evaluation Committee proposed that Simula be given a rolling contract based on evaluations every fifth year and new 10-year contracts contingent upon favourable evaluations. Given that a new five-year contract could be perceived as short-term and engender the risk of researchers leaving as the year 2010 approaches, the administration started to work towards a new contract with the Research Council at an early stage and has made efforts to secure more long-term funding. The Storting's Standing Committee for Finance and Economic Affairs has given a positive signal across party lines regarding a contract up to 2015. This is subject inter alia to Simula undergoing a further international evaluation by the end of 2009. If that also proves favourable, the Ministry of Education and Research will agree to extend the contract with Simula until 2015.

Disagreements among the owners

Even though the report of the Evaluation Committee in 2004 was highly positive, the committee also pointed out some of the challenges facing Simula. One of these was the composition of the board of directors. The Committee recommended that there should be a more equal representation of academia and industry among the board members. The Board consisted of ten members, of whom four were appointed from the Universities, one from the Research Council, one from SINTEF and one from the NR, as well as three representatives of the staff at Simula. The chair of the board at the time, Berit Svendsen, wrote in a letter[7] to the Ministry of Education and Research that she wished to address the recommendations of the Evaluation Committee by reducing the number of board members and changing its composition. Strong disagreements about the future strategy of Simula had arisen among the owners' representatives on the Board. In her letter, Svendsen stated that according to the Limited Liability Companies Act, board members are obliged to work in the best interests of Simula. The disagreements arose from differing interpretations of the exact nature of Simula's mandate and in particular the extent of its collaboration with industry.

[7] Pursuant to The Freedom of Information Act, documents in the public administration are available for the general public.

This was brought to a head in the autumn of 2004 when Simula carried out preliminary work for the industrial giant Hydro[8], resulting in a proposal for a five-year collaboration within the field of computational geoscience. An industrial agreement of such proportions was an excellent opportunity for Simula and was presented to the Board for approval. A conflict ensued between Simula and two of the owners about the collaboration with Hydro, and led to disagreements about the extension of the contract with the Research Council beyond 2010. The research institutes, NR and SINTEF, felt that Simula could develop into a competitor benefiting from far better conditions than they had. This happened at a time when the universities were receiving stronger mandates for innovation work; the Centres of Excellence were being established at the universities, and some of them aimed at stronger connections to industry than the universities had traditionally enjoyed. Together these factors presented a considerable challenge to the research institutes. NR and SINTEF expressed their concerns in letters to the Ministry. In their opinion, the authorities had not sufficiently considered how the centre would fit in with the rest of the Norwegian research establishment. The message was blunt; they wanted Simula closed down and did not support the extension of the contract with the Research Council beyond 2005[9].

However, at the Annual General Meeting on 3 June 2005 the representative of the Ministry of Education and Research, Rolf L. Larsen[10], announced that the Ministry took a positive view of Simula's collaboration with Hydro. He stated that the research collaboration was in line with Simula's mandate as expressed in the Articles of Association, and the Ministry encouraged Simula to continue to seek similar agreements that would contribute to innovation in industry and society. Larsen also pointed out that the company already had substantial public funding, and that such funding could not subsidise assignments carried out for other principals. The proposal to change the composition of the Simula Board was adopted at the AGM, and the new Board held its first meeting in September 2005. The principle of promoting cooperation with industry had been accepted. The Board decided that the technology-oriented development segment of the Hydro project should be organised under the company Kalkulo AS, which was established in 2006. In this way, there would be a clear and documented division between funding for basic research and funding for commercial technology development.

Outstanding researchers

A research laboratory is only as good as its researchers. For this reason, an efficient recruitment policy is essential. At Simula a constant effort is made to recruit ex-

[8] After the merger between Hydro and the petroleum company Statoil in October 2007, the partnership with Hydro was continued with the new company StatoilHydro ASA. According to Forbes Magazine's statistics published in April 2009, StatoilHydro is the 53rd largest company in the world.

[9] Letter of 20 May 2005 to the Research Council of Norway from FIFO (*De norske polytekniske forskningsinstituttenes fellesorganisasjon*).

[10] Records of AGM of Simula Research Laboratory 3 June 2005.

tremely promising candidates, and to some extent the recruitment process is based on headhunting. The best researchers are given funds to develop their own activities within the defined research areas.

Simula aims to build a truly international research centre. Its organisational form has been a deciding factor in the way top scientists are hired, and the way the associated research groups are organised. In particular, Simula has been able to act swiftly and decisively when the opportunity to recruit highly qualified researchers has arisen.

During the last few years, outstanding international researchers have been recruited to the three specific subject areas. The internationally renowned French researcher Professor Lionel Briand, whose previous achievements include the establishment of a laboratory for Software Quality Engineering at Carleton University in Ottawa, was a guest researcher with Simula during a sabbatical year from Ottawa. During the year 2007, Simula recruited Briand, who moved to Norway to take up a permanent position as research scientist in the Software Engineering Department. In 2006 the leader of the Resilient Wireless Networks project, Dr. Yan Zhang, joined Simula's Networks and Distributed Systems Department. He is currently the most productive researcher at Simula in terms of number of publications. In 2008, the Dutch researcher Kirsten ten Tusscher joined the Scientific Computing Department, an outstanding addition to Simula with a strong record of productivity and publication. Dr. ten Tusscher is highly respected and recognised in the international scientific community and acts as leader of a research group at Simula.

In 2003 the Research Council of Norway introduced a new type of support for talented young researchers, the Outstanding Young Investigator scheme. The aim of the scheme was to offer talented younger researchers favourable conditions under which to realise their potential and achieve international excellence. The initiative was intended to play a part in producing talented research managers and to enhance the quality of Norwegian research. In June 2004, Simula researcher Joakim Sundnes was selected to participate in the Outstanding Young Investigator scheme. Competing against no fewer than 221 applicants for a total of 26 grants, Sundnes won a grant from the Research Council. The grant was for building a research group based on a project entitled Computing the Mechanics of the Heart.

In 2007, Anders Logg received the Outstanding Young Investigator grant for the project *EMPH: Automation of Error Control with Application to Fluid-Structure Interaction in Biomedicine.* The project will run over four years and will interact closely with another Simula project addressing robust methods for numerical solution of flow problems. As one of the main developers in the multi-institutional software project FEniCS, Logg has made a remarkable impact and gained extraordinary international recognition for a researcher of his age.

In 2007, the prestigious *Journal of Systems and Software* ranked Professor Magne Jørgensen of the Software Engineering department the best researcher in the world in terms of scientific production among scientists engaged in the development of IT systems. Jørgensen was named the most prolific individual researcher in the world, and of 361 research groups and 3 918 researchers, the Software Engineering Department was ranked third in the world in its field. Only two years prior to this, the same journal ranked Jørgensen the best researcher in Europe, and number three in the world.

Simula has achieved a high level of scientific output. It is reasonable to conclude that this is to some extent due to the emphasis on allowing researchers to work full-time on their research. Since Simula's inception, its researchers have authored six books, 264 journal papers, and 425 conference papers or chapters in books, and edited 17 books. A total of 38 PhD theses have been supervised at Simula, and Simula researchers have participated in the creation of seven companies[11].

A centre characterised by excellence

In June 2007, the SAB undertook a thorough review of Simula. The SAB reports were generally very positive about the development of Simula as a whole and about the individual units. The general message is that Simula merits praise for having achieved a strong scientific position in a very short time, and that its development is highly favourable also in the fields of education and innovation. The SAB review drew particular attention to the exceptionally positive trend achieved by the Networks and Distributed Systems Department. The Scientific Computing Department had already been characterised as *excellent* by the evaluation in 2004. According to the SAB, it has made further advances since then. The Software Engineering Department was also highly commended.

In December 2006 the Research Council of Norway announced that a group at Simula was to be assigned status as a Centre of Excellence. The Centres of Excellence scheme is funded by the Research Council. Its purpose is to establish time-limited research centres devoted to targeted research at a high international level. The scheme also aims to further strengthen the quality of Norwegian research. The new CoE, the Center for Biomedical Computing, was to be based in Simula's Scientific Computing Department. The CoE allocation of a total of 75 million kroner over a ten year period was to be used to conduct world-class research in computing and simulating fluid flows, focusing especially on blood flow through the aorta. Armed with this allocation, which brought with it considerable prestige, Simula were able to enter a vital new field of research.

Innovation and research applications

Innovation work is carried out throughout the entire organisation and is a key element in the research departments. A sound balance between basic research and real-world applications is maintained by organising government-funded research at Simula Research Laboratory, and consulting and technologically oriented projects in stand-alone companies or in cooperation with other partners.

The year 2002 saw the creation of a department dedicated to developing new businesses based on the results of Simula's research and to managing Simula's com-

[11] Data as of 1 March 2009.

mercial interests. In 2004, this new Innovation department became the subsidiary Simula Innovation. Simula Innovation's main responsibility is to promote the application of Simula's research results in society, and the foundation of the company was also an instrument for strengthening the internal focus on innovation. Further, the company takes decisions regarding the acquisition, ownership, and management of stakes in companies.

Collaboration with industry

The research collaboration with StatoilHydro stands out as a milestone in terms of funding and activity level, covering the full range of possibilities from education and basic research activities to commercial development of state-of-the-art technology. Since 2004, the initial contact with the oil and gas division in Hydro has developed into a long-term research collaboration in computational geosciences. The cooperation is of great importance to Simula, partly because of the general difficulty in attracting funding from industry for basic research in Norway. StatoilHydro is currently one of Simula's main partners, accounting for more than ten per cent of Simula's total income.

The collaboration with StatoilHydro continues to gather strength and has a defined scope until at least 2014. The partnership covers a wide range of R&D activities, which all are organised in the Computational Geosciences group at Simula. The consulting company, Kalkulo AS, established in 2006 to provide scientific programming services to industry, is involved in the activity as a sub-contractor. Kalkulo manages the development part of the collaboration thus supporting the research performed at Simula. The main goal of the collaboration with StatoilHydro is to strengthen work procedures in oil and gas exploration through new and improved computer-based models describing geological and geophysical processes. To date, two novel research-based technologies of industrial importance have been successfully established through this collaboration.

StatoilHydro has provided substantial financial support to Simula. Such partnerships are very uncommon in Norway, and the fact that Simula is run as a limited company conducting directed research has been crucial to the establishment of this partnership. This is also true of the agreements with Telenor and Sun Microsystems and more recently Det Norske Veritas. The combination of academic research tasks and technology development works both ways, in that development-oriented activities lead to new PhD and postdoctoral research projects, and vice versa. The intimate connection between the axes of research and development is a particular strength in the StatoilHydro-Simula collaboration. This feature is a consequence of the commitment of both companies to long-term research, and the flexibility offered by Simula's organisational form. StatoilHydro and Simula consider the collaboration to be of strategic importance, and both companies plan for a long-term partnership. Simula's work with innovation and the collaboration with industry is described in more detail in part IV–Research Applications.

Educating researchers

The education of students within the designated subject areas is another of Simula's main tasks. Since 2001, a total of 38 students have completed their PhDs at Simula. The Evaluation Committee noted in the report of 2004 that the number of graduate and post-graduate fellows was low compared to the number of academic staff. This situation was primarily due to inadequate funds. Simula has since taken steps to meet the original expectations and secure additional funding for fellowships.

Research education at Simula is undertaken in close partnership with the University of Oslo. A comprehensive collaboration agreement was signed with the UiO in 2005, based on the mutual acknowledgement that the two institutions have concurrent interests in research, teaching and innovation. The agreement formalises the cooperation, describing the respective responsibilities of UiO and Simula.

At its five-year anniversary celebration in 2006, Simula received some very good news from the then Minister of Education and Research, Øystein Djupedal, who announced, in response to a proposal from Simula, that the Ministry would allocate five million kroner for a school of research within the field of ICT. Simula was then free to invite industrial players to buy shares in the company. The result was the establishment of a joint venture in 2007, the Simula School of Research and Innovation AS[12].

Initiated as a result of a recommendation from the previous evaluation, SSRI was founded with the aim of improving and expanding Simula's efforts within research education. This educational unit is organised as a limited company, and supports all levels of education at Simula: masters students, research trainees, PhD students and postdoctoral fellows. SSRI's main goal is to educate masters and doctoral students in areas of relevance to business and industry in close cooperation with several industrial players and the University of Oslo, which formally confers the degrees. The establishment of SSRI has enabled Simula to expand its PhD programme in terms of both quality and scope.

When SSRI was established, selected industrial partners were invited to join as co-owners. In this way, Simula has sought to maintain an active link between its educational goals and its objective of conducting research on topics of importance to industry and society. Whereas PhD candidates and postdoctoral fellows are employed in the school, the supervisors are based in one of the basic research units. SSRI serves to streamline the educational process, so that both students and supervisors are able to concentrate on the scientific challenges. The SSRI's role also includes close monitoring of the relationship between students and supervisors in order to diagnose and remedy potential problems at an early stage. The overall goal is to make sure that the PhD candidates graduate within the set time-frame and deliver high-quality results. This is particularly useful for personnel recruited abroad, for which the Simula School serves as a step into the university system.

The official opening of the Simula School of Research and Innovation took place on 20 August 2007 and was attended by State Secretary Rikke Lind from the Min-

[12] The majority of the shares is owned, and will always be owned, by Simula (56%). The current co-owners are StatoilHydro (21%), the Municipality of Bærum (14%), Telenor (7%), SINTEF (1%), and the Norwegian Computing Center (1%).

istry of Trade and Industry, as well as representatives from Simula's partners. Chapter 31 "Educating Researchers – a Virtue of Necessity" describes the work with research education at Simula in more detail.

A growing organisation

Since Simula was established in 2001, the number of employees has increased from about 40 in 2001 to just over 100 in 2009. The graph in figure 7.1 shows the proportion of the different categories of employees relative to the total number of man-years[13]. Over the years about half of the employees have been PhD, postdoctoral fellows, or trainees, whereas the proportion of researchers and management and support functions has declined. The scientific staff has increased, whereas the relative number of support staff has declined.

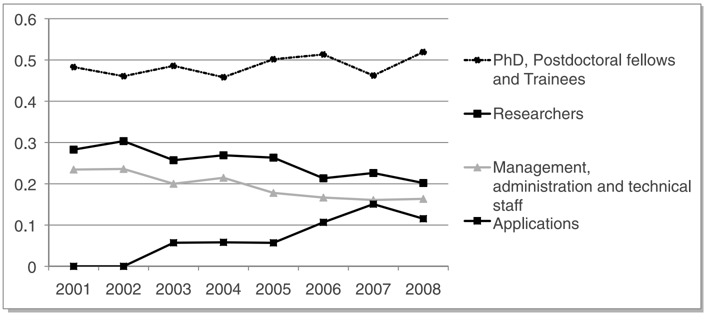


Figure 7.1 Employment Statistics

Simula's original mission was, and continues to be, to conduct basic research in the fields of networks and distributed systems, scientific computing, and software engineering, to promote the application of the research, and to educate researchers in collaboration with Norwegian universities. In the first period, before the evaluation in 2004, the organisation's priority was to release the scientific potential of the research groups. For that reason, the centre consisted of three departments, one for each research group. Over the years, Simula has become a corporation with three subsidiaries. Simula Innovation was established in 2004, Kalkulo in 2006 and SSRI in 2007.

As a result of Simula's rapid growth, it became evident that there was a need to revise Simula's organisational structure. To ensure that management would be able to exercise its strategic responsibility and that routine operations could be dealt

[13] When calculating the number of employees for the purpose of this statistic, man-years have been counted as often as possible and not employees. Also, the numbers indicate the situation at the end of each year. There are, for instance, employees who were full-time during some periods and had temporary positions during others. In such cases, the position's percentage at the end of the year was used.

with efficiently and responsibly every step of the way, the organisation needed an updated structure that would be flexible and consistent with further growth. As of 1 January 2008, the organisational structure was revised to reflect the main tasks of the company, as seen in figure 7.2. Three organisational units, Basic Research, Research Education, and Research Applications, were set up and new heads appointed to run them.

The Basic Research unit deals with the original research activities at Simula and comprises the departments Networks and Distributed Systems, Scientific Computing and Software Engineering.

To highlight the fact that Simula engages in a number of other activities related to innovation and the application of research results, a separate unit was set up with overall responsibility for the application of the research. The Research Applications unit incorporates the two subsidiaries Simula Innovation and Kalkulo. With activities in the same business area, it was considered appropriate for them to comprise a single unit. The organisation allows Simula's innovation work to be viewed in a larger perspective and for activities to be combined with a view to long-term planning and strategic priorities.

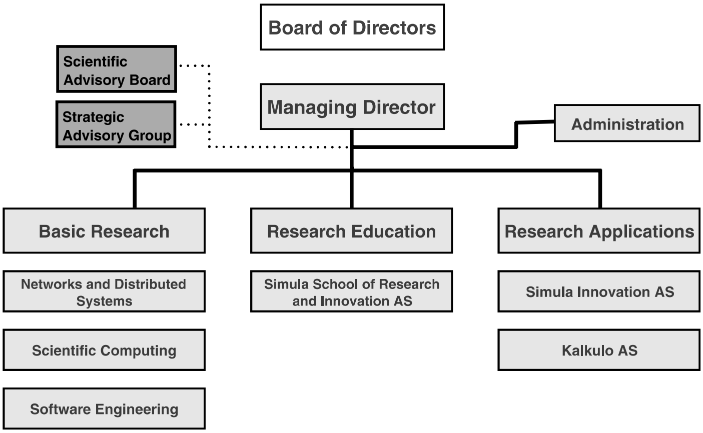


Figure 7.2 Simula's organisation

The Simula School of Research and Innovation was established as an operational unit to deal with all educational activities at Simula. The company is organised under the unit Research Education. Employees pursuing an educational or research qualification are formally affiliated to SSRI, but their research activities are linked to the relevant research department. SSRI takes administrative management of the trainee, doctoral and post doctoral periods, particularly focusing at securing high-quality and relevant follow-up of individual employees in educational positions. In addition, SSRI offers a portfolio of courses that complements the scientific courses at universities. The offered courses, such as in scientific presentation and writing, are essential for the training of successful researchers independently of whether they are pursuing a career in academia, industry, or the public sector.

Maintaining the focus

Since its establishment in 2001, Simula has grown gradually. Along the way, there have been successes, but also challenges. The further development of the centre is dependent on political decisions that are still to be made. In a speech on the occasion of Simula's fifth anniversary, the then Minister of Education and Research, Øystein Djupedal, gave his view[14] of Simula:

> The story of Simula is a success story that bodes well for the future of Norwegian research in general and of Norwegian ICT research in particular. The potential is huge, not least since ICT is unquestionably the most important research area for trade and industry and accounts for nearly 40 per cent of all R&D expenditure by Norwegian businesses. There are clearly more opportunities to be had in this area and significant potential to think about and ponder over! It is impressive to see what Simula has achieved in its first five years. I would therefore like to congratulate all Simula employees, managers and partners, and wish you luck for the future.

Simula's strategy currently runs to the end of 2015. According to the strategy, Simula aims to be a world leader within its selected research fields by 2010 and to accomplish at least one scientific breakthrough and one major research-based commercial success by 2015. To meet these goals Simula will continue to provide excellent support to highly qualified employees, enabling them to focus full-time on achieving significant results, and will maintain an unwavering focus on the three scientific fields throughout the period. Simula will work to preserve its distinguishing features in order to meet the challenges and opportunities of the future.

[14] The Minister's speech (in Norwegian only) can be found at: www.regjeringen.no/nb/dep/kd/aktuelt/taler_artikler/Kunnskapsministerens-taler-og-artikler/Djupedal/2006/Tale-ved-Simulas-5-ars-jubileum.html?id=445066

8

IT FORNEBU AND THE POLITICAL BATTLE THAT LED TO THE CREATION OF SIMULA

Bjarne Røsjø

Simula Research Laboratory was in many ways born of one of the biggest and bitterest industrial policy debates Norway has ever seen.

The formal grounds for the establishment of Simula came in the autumn of 1999 when the Norwegian Parliament discussed the government budget for 2000 and approved the establishment of a research unit at Fornebu. This unit, which was later given the name Simula Research Laboratory, was to form the research nucleus of a future IT and knowledge centre that had been the subject of investigation and discussion for many years.

But a huge amount had happened before the Parliament reached that decision—so much that the IT Fornebu case, as it became known, was brought before the Parliament as many as nine times. The decision to establish Simula had its origins in a debate that began almost ten years earlier, when the Parliament decided that the international airport at Fornebu west of Oslo should be closed down and replaced by a new, modern airport at Gardermoen, north of Oslo.

Bjarne Røsjø
Freelance science writer and communications consultant

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_8,

In November 1995 Statsbygg[1] commissioned a report on the possible options for the future use of the Fornebu area. The task of drawing up the report went to Norsk Investorforum, a lobby organisation which at the time had about 45 members from among Norway's most influential shipowners, entrepreneurs and investors. Norsk Investorforum's representatives emphasised that theirs was an organisation for industry builders with proven track records, not for stock market speculators or largely unproductive financial traders. Statsbygg received its answer as early as January 1996 when Norsk Investorforum came back with a proposal to create an international IT and knowledge centre on the site of the former airport.

The technological future

If the idea of an IT and knowledge centre at Fornebu is to be attributed to one person, then that person has to be shipowner Fred. Olsen[2]. As early as 7 November 1991 he gave a lecture in Trondheim that was given in-depth coverage in one of Norway's main newspapers the following day under the headline "Norges nye veiviser" ("Norway's new guide"). The central thesis of Olsen's talk was that Norway, like other industrialised countries, had entered a late industrial phase in which traditional industries such as the car industry, the aircraft industry and the ship building industry had reached their saturation point. The future lay instead with the new industries such as IT and biotechnology, and if Norway was to be able to enjoy its share of the opportunities–and avoid the pitfalls–that were to result from this paradigm shift, it was time for it to begin the transition to the new economy.

Between 1991 and 1994, one of the people with whom Fred. Olsen discussed his vision was Per Morten Vigtel[3], who, at the time, was head of the Norwegian maritime industry's lobby organisation, the Maritime Forum of Norway. Both men

[1] Statsbygg is a Norwegian government-run enterprise. It acts as key adviser to the government on construction and property issues and owns, manages and develops property on behalf of the Norwegian government. Statsbygg had been given formal responsibility for the subsequent use of Fornebu, and was due to report to an interministerial committee.

[2] Thomas Frederik Olsen, better known as Fred. Olsen, (born 1929), Norwegian shipowner and investor. Fred. Olsen is a visionary businessman, who, as early as the 1960s, recognised the huge opportunities presented by the oil and offshore sectors. Amongst other things, he was chair of the board and a large shareholder in the Aker Group, which was for a time Norway's largest industrial employer. Olsen also had a lot of international business activities including the well-known watch factory Timex in the Philippines.

Fred. Olsen has always been a controversial figure. A story still circulates that Olsen provided the inspiration for the villainous and scrawny capitalist C. Montgomery Burns in the TV series *The Simpsons.* The creators of the series have on several occasions stated that there is nothing behind this story and that any physical likeness is coincidental.

[3] Per Morten Vigtel (born 1941) began his career in *Norges Industriforbund* (the former Association of Norwegian Industry, later consolidated into the Confederation of Norwegian Enterprise (NHO).) He then went on to the Norwegian Shipowners' Association where he was the architect behind the establishment of the Maritime Forum of Norway, a powerful lobby organisation for the maritime industry. Vigtel was also head of the secretariat of Norsk Investorforum. Since 2000 Vigtel has been involved in the establishment of two further lobby organisations, one for the travel industry (*Forum for Reiseliv*) and one for environmental technology (*Forum for Miljøteknologi*).

were, of course, aware that the Storting had approved the closure of the airport at Fornebu, and it was natural for them to suppose that the now vacant airport area with its excellent location could be used to develop a new IT and knowledge centre, which could play a part in bringing Norway into the new era.

When, in November 1995, Statsbygg asked Norsk Investorforum for a proposal on the future use of the former airport area they already had an idea of what sort of proposal would be forthcoming. Norsk Investorforum had already come a long way in developing the idea. During the autumn of 1995 it held meetings with several politicians, among them the Norwegian Labour Party's parliamentary leader Torbjørn Jagland. In October of that year Norsk Investorforum's representatives Håkon Gundersen, Per Morten Vigtel and Fabio Manzetti presented Jagland with a ten page report outlining their ideas for a future knowledge centre at Fornebu.

At this stage Norsk Investorforum's plans were well received. In April 1996 at a meeting at which their plans were presented to an interministerial group one sceptical department director seized the opportunity to ask if there were any private investors willing to put money into the project. Or was the State expected to pay for the whole thing? That was a good question.

Fred. Olsen wanted to return home

A few weeks later the management of Norsk Investorforum had a meeting at the office of Fred. Olsen and the shipowner took the opportunity to contemplate the collection of family portraits hanging on the wall. The foundation for his worldwide business had been laid in 1849, when the British Navigation Acts were repealed leading to a growth in international trade and shipping. The three brothers Fredrik Christian, Petter and Andreas Olsen from Hvitsten, south of Oslo, quickly understood the implications of the opportunities now open to the Norwegian shipping industry. They all became shipmasters, and sailing ship owners with business operations that in time became so extensive that Hvitsten was given its own customs post. "All my ancestors were industry builders in Norway. I was the first generation to leave," said Fred. Olsen.

Already 20 years prior to this Fred. Olsen had built a futuristic technology park by the name of New Tech Park in Singapore, but over the course of the next few weeks he sold off his stake. Now he wanted to so something in Norway. The architect who had designed New Tech Park was now charged with making a drawing of an even more futuristic centre at Fornebu. The architect came back promptly with a project entitled "Sun City Fornebu", largely made of glass, with futuristic lines and with solar heating an important part of the concept. The name was quickly abandoned because "Sun City" had unfortunate associations with a disreputable place of entertainment in the then apartheid state of South Africa, but the idea itself was well received.

In June 1996 Fred. Olsen and Per Morten Vigtel had a meeting with Tormod Hermansen, then CEO of the telecommunications group Telenor. Telenor had plans to build a new head office and it was essential to the success of IT Fornebu that the country's largest IT company moved to Fornebu: it would help to create the beginnings of an IT community and make it more attractive for other companies to establish themselves out there. During the autumn the company decided to set up a new head office at Fornebu with some 6000 employees. The new Telenor headquarters was formally opened in September 2002.

Norsk Investorforum had a meeting with Statsbygg on 9 October 1996, and this time there was a discussion about how much money Fred. Olsen was prepared to invest. Olsen's starting point was that the current property at Fornebu was worth about one billion Norwegian kroner and a letter of intent was drawn up which stipulated that Olsen and the State should each go into the project with 500 million NOK. The letter of intent was written in English by John Wallace, head of Olsen's international business activities.

IT Fornebu was founded

The IT Fornebu company was subsequently founded on 24 October 1996, with Fred. Olsen as chair of the board and Wallace and Vigtel as board members. At this stage there had been no significant debate about the plans. The new organisation set about finding other investors to be co-owners, and after a few months the car company MøllerGruppen, the property company Selvaag AS, the shipping company Anders Wilhelmsen & Co., Norway's largest banking and financial services group Den norske Bank, Telenor, and the shipping-dominated investment company Umoe AS had gone in as joint owners. In addition, BI Norwegian School of Management, the Norwegian Confederation of Trade Unions (LO), the classification society Det Norske Veritas and the technology investor and entrepreneur Terje Mikalsen had also signed up to the vision. Mikalsen had been involved in the establishment in 1967 of Norsk Data, a Norwegian producer of mini computers which at the time were considered among the best in the world.

Norsk Investorforum had taken on the economist Erik S. Reinert[4] as head of research and ideology. The diverse ownership of IT Fornebu was something he was to talk about a lot in the time to come: when an industrial society is replaced by a knowledge society the old dividing lines between work and capital are wiped out. LO is Norway's largest workers' organisation. It maintains close ties with the Norwegian Labour Party and has defended the values of the working classes since its establishment in 1899. BI Norwegian School of Management is more a breeding ground for economists and business leaders–so what were they doing in bed with the capitalists of Norsk Investorforum?

"In the new economy the difference between employers and employees becomes blurred. It is no longer the case that employers own the means of production and that the employees have to toil and slave to generate profits for the capitalists. In a

[4] Erik S. Reinert (born 1949) is a Norwegian economist, social commentator and former business leader. He specialises in development economics and economic history. In 2000 Reinert founded The Other Canon, a network that defines itself as an alternative to the mainstream economic community. Since 2005 he has been Professor at Tallinn University of Technology in Tallinn, Estonia. He is the only Norwegian to be mentioned in Robert L. Heilbroner's book *The Worldly Philosophers*, the world's most widely sold book on the history of economic thought. Reinert's book *How Rich Countries Got Rich and Why Poor Countries Stay Poor* (Constable 2007) was on the Financial Times' bestseller list.

knowledge society the workers take the means of production, in other words their expertise, home with them every day after working hours," said Reinert.

A long time had passed since the European revolutions of 1968, when an industrial policy alliance between capitalists and trade union bosses would have been unthinkable. Erik Reinert was known to quote Karl Marx, but quoted the conservative Austrian-American social economist Joseph Schumpeter (1883–1950) even more often. Schumpeter had emphasised what he called the role of entrepreneurs and innovation in economic development and dismissed the traditional economic theory of "the invisible hand" of the market which was supposed to create a balance between supply and demand. In the opinion of Schumpeter, and Reinert, and Norsk Investorforum such theories did not lead to development and growth.

"We are working together to make the cake bigger, and so we do argue a bit now and then about how the cake is to be divided between us," said LO's vice president Jan Balstad to the daily business newspaper *Dagens Næringsliv.*

Storm clouds gathering

In the autumn of 1996 Thorbjørn Jagland took over as prime minister, following the unexpected resignation of the "mother of the country", Gro Harlem Brundtland. In December 1996 IT Fornebu had a meeting with the then Minister of Planning Bendik Rugaas, who informed them that the government would work to develop a knowledge centre at Fornebu. In the course of 1997 the knowledge centre at Fornebu was included in the government's long-term programme for 1998–2001. It was still the honeymoon period, but the storm clouds had begun to gather.

At that time Prime Minister Jagland made what in retrospect has been described as a major blunder: he announced that he would resign after the October 1997 general election if the Labour Party received less than 36.9 per cent of the vote, the per centage they had gained in the election four years earlier. In the 1997 general election the Labour Party received 35 per cent of the vote and even though the Labour Party was both the largest party and would have been able to form a government, Jagland chose to step down. This cleared the way for a centrist government consisting of the Norwegian Center Party, the Liberal Party of Norway, and the Norwegian Christian Democratic Party led by Kjell Magne Bondevik[5]. The Liberal Party's colourful leader Lars Sponheim took over as Minister of Trade and Industry. He was not at all happy about the plans being made for the use of the former airport site at Fornebu. No sooner had he taken over the ministry from the Labour Party's Grete Knudsen on 17 October 1997 than he announced that not one krone of state funding would go to the IT centre at Fornebu.

In the journal *Plan* (No. 1, 2005) Erik Reinert published a retrospective article with the title "IT Fornebu–næringspolitikk og skiftende virkelighetsforståelse" ("IT

[5] Kjell Magne Bondevik was Prime Minister during the periods 1997–2000 and 2001–2005. Before that he was Minister of Foreign Affairs (1989–1990) and Minister of Church and Education (1983–1986). In January 2006 Bondevik founded the Oslo Center for Peace and Human Rights or "The Oslo Center", which was officially opened on 31 August of that year.

Fornebu–Industrial Policy and Shifting Conceptions of Reality"). In it he describes how IT Fornebu's representatives returned to the office stunned after a meeting with Sponheim on 9 December 1997. Sponheim had started the meeting without greeting them and proceeded to refer to Terje Mikalsen and Fred. Olsen as "obscurantists".

The first big argument about the former airport site at Fornebu had been about how to use the area most effectively. The Municipality of Bærum's head of planning for Fornebu during the period 1992–2001, Hans Kristian Lingsom, describes in the book *Kampen om Fornebu*[6] ("The battle for Fornebu", Norwegian only) how the municipality ended up in an intense tug-of-war with Statsbygg. The head of the Fornebu project for Statsbygg strongly criticised the Fornebu plans in Dagens Næringsliv for their inadequate management and lack of vision, and argued that Bærum was planning to turn Fornebu into a rural area insufficiently exploiting the business area and allowing for too much green space.

The battle over IT Fornebu also contained elements of a more traditional form of conflict in Norway, between the centre and the periphery. Influential groups in Norway's "technological capital" Trondheim did not at all like the idea of having a competitor in Oslo. Opposition also came from technology groups in Kjeller north of Oslo. The Conservative Party's spokesman on industrial policy Ansgar Gabrielsen was a notable opponent. In Lingsom's assessment it was no surprise that Gabrielsen, the most business and industry oriented politician in Southern Norway, became involved in the crusade against Fred. Olsen's vision of an IT centre at Fornebu. According to Lingsom the mobile giant LM Ericsson had more than hinted that the company might establish itself at Fornebu. This would affect its business on the South coast with its head office in Arendal, where a threat from Fornebu was now apparently detected. Nothing ever came of Ericsson's move to Fornebu.

The new centrist minority government was extremely critical of what some described as "fixing." The Labour Party had been in government in Norway for the greater part of the post-World War II period and the non-socialist parties were extremely tired of a political culture where "powerful people talked together and agreed on things behind the scenes, without any kind of democratic checks and balances". Against this backdrop IT Fornebu came to symbolise a political practice with which the Liberal Party and the centrist government fundamentally disagreed. But Sponheim was a minister in a minority government and in this matter the majority of the Storting backed IT Fornebu. The first discussion of the case by the Storting came just after this. The majority of the Storting supported the establishment of an IT and knowledge centre at Fornebu with the State as co-owner and with a strategy for regional spin-off effects.

Christmas dinner interrupted

The political battle reached new heights just before Christmas in 1997. Norsk Investorforum had organised a Christmas dinner at *Det blå kjøkken* the restaurant of the master chef Hroar Dege at Solli plass in Oslo. Per Morten Vigtel had just taken

[6] *Kampen om Fornebu,* Hans Kristian Lingsom, Dinamo Forlag, 2008.

his place at the table when the television news programme *Dagsrevyen* rang. "If you can be outside the Storting before 7pm, you will be the lead news story broadcast live from the Storting!" Vigtel was good at running and was in place outside the Storting at five to seven (6.55 pm) and at 7 o'clock the Norwegian people learned that "a large majority of the Storting had voted in favour of establishing a knowledge centre at Fornebu." All parties of the Storting, with the exception of the Socialist Left Party of Norway, now supported the idea of a centre at Fornebu. Instead, the disagreement was now about the size and make-up of the IT centre and who would be allowed to run it.

In 1996 the Conservative Party had supported Norsk Investorforum's plans, while the Progress Party[7] had been against them. But during the course of 1998 the two parties swapped sides. Progress Party leader Carl I. Hagen had understood that the traditional differences between right and left in Norwegian society were on the point of weakening, thus creating new opportunities for the party. The result was that former arch enemies the Labour Party and the Progress Party now had the pleasure of working together to realise the IT Fornebu project. The Labour Party put the taciturn but extremely strong-willed former Minister of Transport and Communications and former Minister of Local Government and Regional Development Kjell Opseth on the case, while the Progress Party designated the equally controversial Øystein Hedstrøm as spokesman on industrial policy. Hedstrøm was a member of the Storting from 1989 to 2005 and made his mark, amongst other things, as a powerful spokesman for a stricter Norwegian immigration policy.

Europe's most attractive knowledge centre

In an interview with *Dagens Næringsliv* in 1998 Per Morten Vigtel described his vision of making IT Fornebu the most attractive knowledge centre in Europe by the year 2005. At the time Erik Reinert, Per Morten Vigtel and IT Fornebu's project manager Gudmund Hernes[8] were travelling all over the country spreading the good news about a knowledge centre that could transform Norway into a knowledge society.

"Norway stands at the threshold of a transition to a new technical-financial age. This transition is so significant that it warrants being called a paradigm shift, that is, a fundamental transition from one technological standard to another. We are on the point of leaving the age of mass production and entering the information and communication age. In the previous era, oil played a key role as a new and cheap

[7] The Progress Party was founded in 1973 as a liberalist party, but is also described as populist and has made large inroads into the voting base of several other parties.

[8] Gudmund Hernes (born 1941) is a sociologist, writer, former politician and former government minister for the Labour Party. In the autumn of 1990 Hernes became Minster of Education, Research and Church Affairs in Gro Harlem Brundtland's third government, a position he held until he became Minister of Health in the same government from December 1995 to October 1997.

During the period 1997–1999 Hernes was involved with IT Fornebu and Norsk Investorforum. In December 1999 he became Director of UNESCO's International Institute for Educational Planning.

Since 2006 he has returned to research at the research foundation Fafo and is Professor II at the Department of Public Governance, BI Norwegian School of Management.

resource, but in the coming era micro electronics will be the new driving force," said Reinert.

The shipowner's "clammy, grasping hands"

By this stage the debate had become extremely heated, and not all arguments were equally objective. IT Fornebu was accused of wanting to acquire "the Norwegian property market's tenderloin steak" and of wanting to "siphon the finances out of the public purse." At the beginning of April 1999 Fred. Olsen telephoned the author of this narrative[9], at the time a journalist and research associate at *Dagens Næringsliv,* from his holiday resort at La Gomera in the Canary Islands. It caused something of a stir at the paper because Olsen was not someone to offer his opinions freely to the press. But now he had been accused by the Conservative Party and the Socialist Left Party of "fixing" with the Labour Party and the Progress Party, and the business journal *Kapital* had recently accused him of having "clammy hands, just waiting to get hold of as many of the country's most valuable plots of land as possible."

Olsen thought the criticism was unfair and it was clear to me that he was both sincere and annoyed. The telephone conversation developed into an interview in which Olsen told me that Progress Party leader Carl I. Hagen had *not* been collected by limousine and served a luxurious meal at his premises. Newspaper reports had cited this as the explanation for Hagen's change of attitude and new positive stance towards IT Fornebu. "What nonsense. We had a meeting with Hagen at IT Fornebu's offices right next to the Storting, and we both managed to get there under our own steam. Hagen may have been given a bread roll or two," Olsen explained.

Fred. Olsen was also extremely critical of those who claimed that the State should stay well away from new business development. "There are a lot of people claiming that Silicon Valley was established by private business, but nothing is further from the truth. The American IT success story began with Fairchild Industries, Stanford University and Hewlett Packard receiving funds from the military development programmes. This was developed into NASA and Kennedy's moon programme. Now we need to create a small moon programme for Norway, in the sense that the public sector organisations really enter the brain revolution," Olsen insisted.

The journal *Kapital* had picked up on the fact that Olsen did not use a computer himself, and made a big issue out of the fact that "now he plans to build an IT centre, a man who thinks that surfing is something you do on the beaches of California." When I asked for a comment, Olsen's initial answer was rather unclear, but he rang back half an hour after the interview was over and explained why: he had never learned to write correctly. The man was dyslexic and needed help with anything that

[9] If ten people were to write the history of IT Fornebu, the result would probably be at least five completely different versions. The author was a research journalist at Dagens Næringsliv and covered the IT Fornebu debate during the period 1997–1999. This historical review is written against the background of my own articles as well as other newspaper clippings from the time, together with additional conversations with a few of the central figures. I would have liked to have talked to a number of other key individuals, but the result would then have been a book instead of a chapter.

had to be written down. This is incidentally a condition that Olsen shares with a number of other successful Norwegian business people and industry builders.

In his autobiography *Ærlig talt*[10] ("My Memoirs", Norwegian only), Carl I. Hagen describes how he came to change his opinion after the meeting with Fred. Olsen. In Hagen's account, Fred. Olsen exuded a passion for his country and a commitment to playing a part in and developing Norwegian society. Fred. Olsen, then many years Hagen's senior, explained that the key to success lay in creating a good environment with a large number of people in different companies who got to know each other also outside work.

The decision approaches

Minister of Trade and Industry Sponheim was still against IT Fornebu being given the project because "a few people had talked together". He wanted to put things on a regular footing by opening the job up to international competition. By the deadline in May 1999 six interested parties had come forward. Not all of them were equally genuine, but after an initial sifting three serious competitors were left on the battlefield. One of IT Fornebu's new rivals was called Fornebu Technoport and was a collaboration initiative between Oslo Innovation Center and KLP Eiendom[11], with the former Deputy Director General of NHO (Confederation of Norwegian Enterprise) Gro Brækken as project manager. The other competitor was Nettverk Fornebu, a cooperation project between the industry conglomerate Orkla, the insurance giant Storebrand and the wholesale and retail group Hakon Gruppen. The key people were then CEO of the retail group Steen & Strøm Erik Bøhler, the investor and "food king" Stein Erik Hagen and CEO of Orkla Halvor Stenstadvold.

IT Fornebu had received the distinct impression that the government wanted to avoid giving them the job if possible, and in August 1999 it was announced that IT Fornebu and Fornebu Technoport had merged. The new organisation was called IT Fornebu Technoport, and many people believed its establishment effectively meant the sidelining of Nettverk Fornebu. The plans of the new organisation focused primarily on the development of 31.6 hectares that had been put up for sale by the State for the establishment of a knowledge centre, but the partners also wanted to buy up an additional area bringing them up to a total of some 55 hectares. There was talk of establishing a centre with some 8000 employees, of which 4000 were to be researchers.

The Ministry of Trade and Industry appointed a committee to review the proposals of the two competitors and to recommend a winner. The committee consisted mainly of lawyers and representatives of Statsbygg together with two academic experts, namely the Rector of the Norwegian University of Science and Technology Emil Spjøtvoll and IT expert Professor Morten Dæhlen, who at the time was Executive Director of Science and Technology at the Research Council of Norway. Dæhlen

[10] *Ærlig Talt*, Carl I.Hagen, Cappelen Damm, 2007.

[11] KLP Eiendom AS: One of Norway's largest property companies and a wholly owned subsidiary of Kommunal Landspensjonskasse (KLP), a mutual insurance company.

and Spjøtvoll quickly agreed that Nettverk Fornebu had come up with the best proposal and for a long time it looked as if their view would be supported by the majority of the committee. However, in the event the rest of the committee, i.e. the majority, chose to back IT Fornebu. This meant that IT Fornebu's supporters could breathe a sigh of relief, because it would have been extremely difficult for the Storting to go against a recommendation from this committee.

An unexpected telephone call

Where did the idea come from that the IT and knowledge centre should include a state funded institute dedicated to basic research? It was certainly not a part of the IT Fornebu Technoport proposal. The idea for what would later become the Simula Research Laboratory was first aired in public in connection with the discussion of the State budget for 2000, when the Storting approved the establishment of a research unit at Fornebu.

Today it can be revealed that the idea came in all probability directly from the then Minister of Education, Research and Church Affairs Jon Lilletun (1945–2006). Morten Dæhlen was sitting in his office at the Research Council one autumn day in 1999 when he received an unexpected telephone call from an administrator at Lilletun's ministry with a crystal clear message from the minister: if there is to be an IT and knowledge centre at Fornebu, a basic research centre should also be established that can be its research nucleus... The Research Council was told to set to work coordinating the planning and establishment of the research unit as quickly as possible exploring the options for setting up such a research centre. The basic research centre was to be established at Fornebu, independent of whether the development project went to IT Fornebu Technoport or Nettverk Fornebu.

Morten Dæhlen later had the opportunity to ask Lilletun directly where the idea came from and he was left with the clear impression that the initiative was the minister's own brainchild. The Norwegian Christian Democratic Party politician Jon Lilletun—a man who only had an elementary education—has moreover gone down in history as one of Norway's most competent and committed ministers responsible for research. Everybody knows that government ministers who want more money allocated to their ministries have to fight their corner hard with the Ministry of Finance, but those who had the opportunity to meet Lilletun were quickly convinced that he was prepared to do anything in his power to strengthen Norwegian research.

The final decision

The final decision on the case came on 7 March 2000, when the Storting chose IT Fornebu Technoport as the State's partner in the development of an IT, knowledge and innovation centre at Fornebu. Carl I. Hagen also refers to this in his memoirs. He recalls that the Minister of Trade and Industry Lars Sponheim was beside himself with rage because he had been outvoted and because others were seeking a positive business development. The Progress Party on the other hand had been pleased. As well as choosing the best and most ambitious alternative, they had also had the pleasure of thumbing their noses at Per Kristian Foss and the Conservative Party, who were completely against the decision. In Hagen's version, the Conservative Party's bitter fight against IT Fornebu was motivated by their opinion that the State should not bear any responsibility for the nation's development and should not step in to facilitate business development.

At the end of the year EFTA's surveillance authority (ESA) approved the system of contracts with the State and everything was ready to go. But by this stage the so-called dot-com bubble was about to burst and during the course of 2001 it became very difficult to attract investors to IT projects. The result was that the knowledge park at Fornebu got off to a delayed and slow start.

What became of the vision

In terms of substance, the original and ambitious visions of an international knowledge centre at Fornebu have so far not been realised. Some Simula researchers say that as far as they were concerned the vision shattered as early as 2001, when they were due to move into their new, smart premises. The management of Simula inspected the site on the Friday before they were due to move in and everything seemed in order. But three days later, when the researchers arrived with removals vans full of digital wares they were met with a scene of total chaos. The builders continued to work for several months, the key cards did not work, and the canteen was only temporary. The parking wardens, on the other hand, were as efficient as vultures and handed out fines to an alarmingly high number of visitors.

In February 2004 the Danish consultancy Oxford Research AS delivered an evaluation of the IT Fornebu initiative. The evaluation showed, amongst other things, that the Fornebu project around Simula had not developed as planned and it was recommended that the project be revitalised during the course of 2005. The then Minister of Trade and Industry, Ansgar Gabrielsen, thanked Oxford Research for the report and promptly put the revitalisation plans in his desk drawer.

But periods of decline do not usually last for ever, and after a while the tenants began to drift in to Fornebu. At the turn of the year 2008–2009 IT Fornebu Eiendom had tenants with a total of approximately 2200 employees and the company had plans for further rapid and extensive expansion. Already long before this, Telenor had established its international head office at Fornebu with over 6000 employees. The head office of the paper producer Norske Skog was situated on the other side of the Fornebu peninsula. The industrial group Aker Solutions, which has some 26 000 employees in 30 countries, moved into its new head office at Fornebu in 2007 and Hewlett-Packard Norge was due to move into IT Fornebu's new portal building in 2009. In February 2009 the international energy company StatoilHydro ASA approved a decision to move its approximately 2000 employees in the Oslo area to IT Fornebu in 2012. The whole Fornebu area would therefore encompass more than 12 000 "knowledge workers" if you count both IT Fornebu's own tenants and the other companies.

In retrospect it can be concluded that the debate about IT Fornebu contributed to putting both information technology and the knowledge society on the agenda in Norway, and did result in the creation of a centre at Fornebu. The centre did not turn out completely as expected in terms of its substance, but the vision regarding its size–approximately 15,000 new jobs at Fornebu by 2015–looked set to be fulfilled. The planned development of residential accommodation, however, was something of a tragedy. But that is another story.

9

THE RIGHT STEP AT THE RIGHT TIME

An interview with Paul Chaffey by Bjarne Røsjø

"Setting up Simula was the right step at the right time for Norwegian ICT competence" says Paul Chaffey, Managing Director of Abelia, the Business Association of Norwegian knowledge- and technology-based companies.

"Simula is an exciting model that could also be relevant to a few other areas of the Norwegian education and research system. I am thinking particularly of biotechnology and nanotechnology, which are cross-disciplinary subjects in the same way as information and communication technology. But I don't think that research centres modelled on Simula should be established across the board," says Chaffey.

Paul Chaffey has been head of Abelia since the organisation was set up in 2001. Abelia[1] works to protect the business interests of its over 700 member companies and is also an employers' organisation affiliated to the Confederation of Norwegian Enterprise (NHO). NHO is an important stakeholder in Norwegian society, in part because each year the organisation conducts pay negotiations with the employers' organisations.

Chaffey has been a public figure in Norway since 1989 when, at the age of 24, he was elected to the Storting as parliamentary representative for the Socialist Left Party of Norway (SV). Chaffey later distanced himself from the party's political views and has instead developed an ever-growing interest in research, innovation and industrial development. Today, amongst other things, he serves as a member of the boards of the University of Oslo and the Norwegian Business and Industry Security Council (NSR).

[1] Abelia is named after Niels Henrik Abel (1802–1829), who is one the most famous Norwegian mathematicians of all time and who is still well known internationally.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_9,

Paul Chaffey

ICT, biotech and nanotech are special areas

Chaffey's reasons for suggesting that the Simula model could have limited applicability are that large parts of Norwegian trade and industry are not driven by basic research in the same way as ICT, biotechnology and nanotechnology. "Companies in the construction, process or transport industries, for example, do not carry out a lot of their own research, but instead base their innovation on the results of other peoples' research. So it is not necessary for all industry-oriented research groups to be fully funded by the State. That is why I believe that we should retain the Simula model for just a few select areas, for which Simula provides an excellent model," says Chaffey.

Some 70–80 000 people are employed by pure ICT companies in Norway. Moreover, ICT pervades most other areas of economic activity, so the total number of qualified ICT professionals or people with ICT responsibility is far higher than this.

Integration needs to be improved

"One of the things that Simula has shown us is how a better integration of research and industry can be achieved. We have to become better at this in Norway, if we are to have a successful industrial policy. As a high cost country we cannot compete internationally when it comes to manufacturing, so we must focus our efforts on creating a knowledge-intensive industrial sector instead. This means that business and industry should seek out research findings both in Norway and abroad to a greater extent than they do today, and should recruit more people with higher education. Such skilled labour resources will be fought over in the years to come," Chaffey explains.

"One problem in Norway has been that the gap between the research and educational institutions on the one hand, and business and industry on the other has been too large," states Chaffey. "Until now we have solved this problem by having a large number of independent research institutes operating somewhere between research and industry, whose focus has largely been on applied research. This is a good system that has created a market dynamic in applied research. But this system has also given rise to a problem in that pure research groups at the universities have lacked exposure to industry and the market. It is precisely here that Simula provides a solution, because it has shown us how the gap between pure research and industry can be bridged. Simula has produced research results that could not so easily have been generated at an applied research institute. This is because the institutes' dependence from an industrial sector that can pay for the findings makes it difficult for them to implement extensive and long-term research projects," he explains.

An exciting development

Paul Chaffey knows Simula well, not least because the centre has been a member of Abelia since its establishment in 2001. He was among the guests at the official opening of the new premises at Fornebu in April 2002, and gave a speech at the centre's five-year anniversary celebration in 2006. "It has been exciting to follow the development of Simula. The centre has not only produced excellent scientific results, but has also managed to promote itself as a success and to create enthusiasm for the work that it does. In this respect Simula could also be used as a model: elsewhere in the academic world in Norway there is not much emphasis on self-promotion. But if you don't say that you are good at what you do yourself, you can be sure that no-one else will."

Chaffey is pleased that Simula has been able to attract a large number of foreign students, but also sees that this has its problematic side. "It is a paradox that the major information and technology study programmes in Norway are attractive internationally, but unfortunate that so few Norwegian young people choose them. It makes us vulnerable, because some of the foreign students will no doubt return home or may go to the USA after they have completed their education. We can only hope that Norwegian industry will improve its track record when it comes to welcoming these skilled individuals and that Norwegian immigration regulations become more relaxed. Today the rules are so inflexible, that we are losing talent that is sought after in the rest of the world."

More qualified ICT professionals needed

"There is a huge shortage of people with an education in ICT among Abelia's member companies and in the rest of the industrial sector," states Chaffey. "We need more people both with shorter and vocational qualifications, and more ICT professionals with long and thorough educational backgrounds. But today the need for this expertise has shifted partly from the ICT branch itself to other companies in general. In the future we will also need more ICT professionals, but it will be increasingly important to be able to combine ICT with other disciplines, such as design or finance. That is to say nothing of the combination between ICT and biotechnology, which is an area with huge potential."

Chaffey thinks it is paradoxical that as consumers of ICT, Norwegians are among the top in the world, but that Norway only plays a small part in developing new products and services. "This is partly because neither the ICT sector nor the educational institutions have been good enough at drawing attention to the opportunities that exist. Simula has also been more successful than most in this respect," he points out.

10

A BRIEF HISTORY OF NORWEGIAN SCIENCE AND RESEARCH POLICY

Christian Hambro

Setting the scene

Writing a brief history of Norwegian science and research policy in a few pages is a daunting task. However, being given the opportunity to do so is both a high honour and an irresistible challenge.

History is always transmitted through the eyes of the author, a fact that limits its objectivity. This is particularly the case when writing about the recent past and right up to the present. The border between history on the one side and commentary on current affairs on the other can, in such cases, easily become blurred. These dangers are particularly obvious when the author has taken part in the recent history, as the director of the Research Council of Norway for a number of years. In addition, the author is in no way a historian. Hereby the readers have been warned.

To underline that the present chapter is not a scholarly exposé, the tone is kept lighter than that which a historian might attempt to convey. Footnotes are absent, in order to avoid any resemblance to a scientific article. However, Magnus Gulbrandsen and Tore Li, who both are specialists in the history of Norwegian research and development (R&D) policies, have kindly read through the manuscript. Their advice has, to a large degree, been followed. Although they in no way are accountable for the chapter, their comments seem to indicate that the story presented is not too far off the mark, and the author is thankful for the improvements they have suggested.

Christian Hambro
Gram, Hambro & Garman, www.ghg.no

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_10,

Although there are no footnotes, it should be mentioned that the chapter is based on the reading of a variety of publications: *Det Kongelige Fredriks Universitet 1811–1911 Festskrift, Universitetet fra 1813–1911* av Bredo Morgenstierne; *Universitetet i Bergens historie*, Astrid Forland og Anders Haaland; "Norsk forskningspolitikk i etterkrigstiden", Hans Skoie, NOU 1981:30 *Forskning, teknisk utvikling og industriell innovasjon*; "Public R&D and Industrial Innovation in Norway: A Historical Perspective" by Magnus Guldbrandsen and Lars Nerdrum in Innovation, Path Dependency, and Policy, (Fagerberg, Mowery, & Verspagen. eds., Oxford University Press); *The Co-evolution of Research Institutes with Universities and User Needs: A Historical Perspective* by Magnus Gulbrandsen. In addition, the chapter is based on several official publications, governmental whitepapers, budget documents, and on subjective interpretations of personal experiences as the director general of RCN.

A history difficult to grasp

The chapter scetches a rough picture of Norwegian R&D policy and places it in a historical perspective. It attempts to give a description of the developments that might be of interest when trying to understand the current climate of Norwegian R&D and R&D policy. The chapter is not a history of the Simula Research Laboratory, although the establishment of Simula has been given some attention. But it is hoped that the chapter does accurately situate Simula as a new type of institution in the evolutionary development of the Norwegian R&D system.

The presentation does not follow a strict chronological line, because there are a number of different themes in the history of Norwegian R&D policy and development that sometimes are only loosely linked and therefore best dealt with separately. The result of this approach is at times a rather disconnected presentation of several themes. But this might in many ways be a fair way of telling the story, since Norwegian R&D policy has never been crystal clear or followed a straight line. In modern times, it can best be compared to a kaleidoscopic pattern that is changing continuously, and is therefore difficult to grasp. The overall picture, which also indicates the themes dealt with later on in this chapter, is outlined below.

Between 1945 and extending up to the 1960s is a period that can be characterized by an instrumental belief in science and R&D as a tool for developing both the economy and society in Norway. This then was a period of directed R&D growth. The period from the mid-60s up to the mid-80s was the epoch of student-driven expansion of the Norwegian R&D system. The period from the beginning of the 1990s and continuing to the present has been one of continued expansion, with increased attention given to quality in science and internationalization of Norwegian R&D. The number of research institutions grew considerably in the period 1950–1970, creating an increasingly fragmented Norwegian R&D system, probably at the cost of scientific excellence, but beneficial for society and the regions. Increased government involvement in R&D matters necessitated the establishment of research councils. These have generally had a troublesome relationship amidst the scientific community, the business sector, and the ministries, in part due to the tasks they have been given, weak governmental coordination of R&D policy, and the tendency many ministries have had to micromanage R&D funds that have been channelled through the research council system. Overall, R&D policy has, except for a few years after World War II, never been regarded as a particularly important policy area, either by the public at large or by politicians. Support for science, as part of culture, has traditionally been weak in Norway. On the other hand, history has shown that there has been solid public support for R&D projects that have been perceived to be of practical importance. Thus, Norwegian R&D, historically and up to the present day, has been particularly successful when related to natural resources and businesses with long traditions in Norway. Norwegian R&D policy and the organization of the research system has usually followed a pragmatic step-by-step approach, more characteristic of an evolutionary system than any grand design. The result is a system that is more flexible and attuned to national and regional needs than one characterized by a zeal for scientific excellence. Today's situation in Norwegian R&D and R&D policy thus can be traced backwards in time and is but one of many examples of path dependency in national development.

The establishment of Norway's first university

Research, and in particular development, is nothing new in Norway. Viking ships were the floating high technology of their time, and much could be said about the development of these fascinating vessels. However, in terms of the history of R&D policy, it seems more reasonable to choose events closer in time as our point of departure.

Norway is a latecomer to the party of tertiary academia. The University of Oslo was founded in 1811, whereas the oldest European universities were created more than 500 years earlier. Yet, this comparison is not entirely fair, since the University of Copenhagen was established in 1479, and Denmark and Norway were united from 1387 to 1814.

The initial reasons for proposing a university in Norway were to mitigate the costs and dangers of sending young people to study in sinful Copenhagen, without parental supervision. It was later admitted, with much insight, that youth might also be tempted to commit sins in towns smaller than Copenhagen, even in Oslo. Gradually, the arguments changed and became oriented in the direction of a university's value to society. A number of university initiatives were undertaken in the late 18th century and in the beginning of the 19th century, to no avail. The suggestions were not linked to the idea of an independent Norwegian nation, although the king saw the issue in such a context.

King Fredrik VI was ambivalent regarding the establishment of a Norwegian university, although he did acknowledge the burden of sending young people to a large town in another country for many years. As a compromise, he suggested appointing four professors in each of the seven dioceses who could offer basic university courses in Norway. The educational administration in Copenhagen stated that finding 28 capable men for the task was an insurmountable challenge. While this may sound incredible, the truth in this belief was later, to some degree, confirmed when the UiO started recruiting professors. One should note that Norway's population at the time, the early 1800s, was a scant 885 000 inhabitants, and only ten per cent lived in urban areas. The king's administration presented an alternative solution by proposing a new educational institution in the small city of Kongsberg, instead of a university. This, then, was the king's decision in 1810.

His royal decree was, however, not implemented. International developments linked to the Napoleonic wars and the sudden death of the Swedish Crown Prince Kristian August, resulted in an unstable situation for the Danish king. He was, quite rightly, now concerned that influential Norwegians might desire Norway's independence. The desire for a Norwegian university was stronger than ever and had become an issue related to Norway as a nation. To please the Norwegian citizens, the King eventually granted permission for the establishment of the University of Oslo in September 1811. This was no great burden for the public purse, as the Norwegians said they would finance the university through private gifts. The subscription was a considerable success, and was closed in 1813, with enough money to set up the necessary buildings.

By 1820, the UiO had appointed some 20 professors. A mere 185 professors were appointed in the whole period extending from 1813 to 1911, an average of 1.9 new professors a year, which might be said to have been only slightly above the rate necessary to keep the university alive. The annual enrolment of students until the 1860s was approximately 100.

Norway has never again witnessed, since the establishment of UiO, the same private generosity, as regards the financing of science. Private donations have not come anywhere near the gifts to science, for example, as those in Denmark, Sweden, England, and the USA. In recent years, however, some very wealthy Norwegians have started to open their purses, and a new, substantial endowment from Norway's "king" of real estate has been announced.

Although underfunded, Norwegian science thrived in the 19th century

The UiO was parsimoniously funded from the very beginning. The number of positions was very limited, there was little money available for research, and there were no laboratory facilities to speak of. Norwegian scientists continue to grumble about much the same today.

Taking into account the meagre financial start and the small population, it is most surprising indeed that Norway turned out a handful of eminent scholars in the 19th and early 20th centuries. In mathematics, Nils Henrik Abel; in astrophysics, Kristian Birkeland; in geology, Waldemar Brøgger, Baltazar Keilhau, and Victor Goldschmidt; in chemistry, Cato Gullberg and Petter Waage; in meterology, Vilhelm Bjerknes; in marine biology, Michael Sars, Georg Sars, and Johan Hjort; and in medicine, Armauer Hansen.

This period was in many respects a golden era for Norway, not least of all culturally. It might seem as if cultural, economic, and scientific success is interlinked in a manner we do not entirely understand. One hypothesis explaining the historic academic success could paradoxically be the lack of public appropriations for university research. In order to obtain funds for their research, professors have always had to cooperate with industry and governmental institutions. And, at any rate at that time, contact with problems at hand seemed to also stimulate scientific creativity. And as there were not any research facilities, the pioneers used nature as their laboratories. It is hard to believe today that meagre funding could bring about great leaps in natural sciences.

One of the most dramatic stories related to science financing is linked to the astrophysicist Kristian Birkeland. In order to obtain funds for his research, he was engaged in a number of projects, and one was the development of an electric cannon that he patented. The demonstration of the cannon to the Norwegian elite was a total flop, with a nearly deafening bang that sent the invited guests out into the streets, panic-stricken. Birkeland, noticing the smell of nitrous gasses after the colossal electrical discharge, immediately grasped the fact that electricity could be used to extract nitrogen from the air. He also knew that more fertilizer was necessary to ensure future agricultural production. Thus was born Norsk Hydro, the largest company in Norway for more than one hundred years and a world leader in the field of artificial fertilizers to this day. Birkeland was a co-funder of Norsk Hydro and earned enough money through the sale of his shares in the company to fund his research. It should, however, be mentioned that the only aim of Birkeland's pursuit of industrial projects was to earn money for his basic research.

In modern times, an excellent example of cross-fertilization between basic research and solving practical problems is the work of Kristen Nygaard and Ole Johan Dahl, who, amongst other honours, were awarded two of the most prestigious prizes in computer science, the Turing Award and the John von Neumann Medal. They were given the prizes for the invention of object-oriented modelling and programming, the most widely used programming model in our time. The Simula language they developed during the 1960s provides the underlying logic for many modern computer languages in use today, including C++ and Java. Their inspiration came from practical tasks related to process control at the Norwegian nuclear reactor at Kjeller. The Simula Research Laboratory, discussed later in this chapter, has taken its name from the programming language, thus honouring these two pioneers who developed the concept.

The poor funding of research and science that began in the 19th century continued up to the end of World War II. Even today, it continues to be regarded as a major problem by the universities. Political understanding of the importance of science in and of itself has never been strong in Norway. Moreover, the population as a whole has shown little interest. And, generally speaking, there has been a preference for using knowledge for incremental improvements, more than cultivating science as a long-term investment. Intellectuals naturally regard this situation as an unhappy one. However, taking into account Norway's considerable wealth today, the high productivity in most fields, and a welfare state to be proud of, it is not certain that much better funding of science and research in the past would have benefitted the nation to any considerable degree. What is called for in the future is another issue.

The belief in knowledge in support of natural resources' use has been strong

In the 19th century, popular support for science as such was limited to a small elite in the few cities of Norway. However, the situation was not entirely bleak from a scientific point of view, because there was a wider understanding of the value of knowledge and education to enhance production and to manage natural resources.

The king's initial decision to establish a new educational institution in Norway in 1810, was not to locate it to Oslo, but was, as already mentioned, to place it in the small city of Kongsberg. The idea was probably based on the presumption that a location close to important economic activities was sensible, and even more so, as there already was a nucleus of an academic institution in Kongsberg.

Kongsberg's silver mines were amongst the largest in Europe, and employment peaked at approximately 4 000 people in 1770. The mines were indeed of great importance to the kingdom of Denmark-Norway. The Mining School in Kongsberg was established as early as 1757, with the task of educating mine managers. The school was one of the first European higher educational institutions in its field and probably the first Norwegian institution with any semblance of an academic approach in its mission.

The Mining School was eventually shut down, and its tasks taken over by the UiO in 1814. The UiO had a rather detached attitude in relation to the practical needs of society, so the transfer from Kongsberg to Oslo was not a complete success. In 1858, the Geological Survey of Norway was established. The combination of an academic tradition and a public agency proved to be very beneficial for the development of geological competence. Norway has held a strong international position in geology up to this day, something that has been of great importance for the Norwegian petroleum sector since the 1970s.

Fishing has historically been the backbone of Norwegian economy and culture. No wonder then that Norwegian scientists, from the very beginning of their academic pursuits, were heavily engaged in marine biology and oceanography. Scientific knowledge was used as early as the 19th century to resolve conflicts between whalers and coastal fishermen. In 1900, the Directorate of Fisheries was established. It consisted of two distinct parts, one dealing with fishery politics and administration, the other with scientific research. The latter was eventually spun out as the independent Institute of Marine Research. Once again, we see a typical development in Norway: Sparse funding of independent research; more generous funding if the science is closely knit to identifiable challenges of national importance. And, as before, it seems that the symbiosis between basic science and applied research has been fruitful. Norway is still very much in the forefront in marine sciences and has had considerable success in the development of the modern fish-farming industry.

In the 19th century and extending long into the 20th century, Norway was also an agricultural nation. The arable land was limited and the climate was not favourable for farming, which led to a considerable exodus of the population from Norway to America. But it also induced a clear understanding of the importance of using modern methods in agriculture, and that agricultural education was necessary. Thus, the College of Agriculture was established in 1859 with the task of teaching "farmers how the upper classes ran their farms". The college gradually developed into a research institution and has achieved status as the Norwegian University of Life Sciences in 2005. The institution has had great importance for the development of Norwegian agriculture, was instrumental for developing the fish-farming industry, and is today an important player in the field of biotechnology.

These examples from the 19th and early 20th century have paved the way for Norwegian science and research policy up to the present: weak popular and political support for science as such, strong belief in knowledge and education, and a willingness to develop knowledge and insight in fields with a perceived practical importance.

The public lack of interest in science itself has so far had a limited impact

As we have seen, political and popular attitudes in the 19th century in some ways carry over to this day, although the rhetoric now is more science-friendly than in the past. The positive aspect of this is a down-to-earth development of the innovation system. The tendency has been to invest in what has been perceived to be useful science, and not taking excessive risks by investing heavily in science as a value in it self. Although this in one sense is laudable, the attitude also entails the risk of leaving the country in the dust while other nations charge forward scientifically.

The industrialization of Norway gathered momentum in the 19th century, and one would have thought that creating a Norwegian institute of technology would make sense, for example modelled after the French École Polytechnique that was founded in 1794, or the Technische Hochschules that were established in German-speaking countries in the last half of the 19th century. But the Parliament turned down the first proposal for such an institute on the grounds that the industrial sector was too small to justify the costs. It should be noted that the Parliament, at the time, was dominated by fishermen, farmers, and merchants, with no particular interest or insight in the industrial revolution that was sweeping over Europe and across the United States.

That line of thinking, prevalent in the 19th century, finds its parallel in today's discussions about research funding. Some say it is reasonable that Norway's R&D effort as a per centage of GDP is low, as the country supposedly is dominated by natural-resource businesses that do not call for much R&D. A more dynamic perspective would be that increased investment in R&D is precisely what is needed to develop a more knowledge-intensive business structure that can secure new sources of future income when the petroleum income dwindles. None of these strains of thought deserve full support. The natural-resource-based industries are very knowledge intensive, but they achieve this without doing much R&D themselves. On the other hand, pouring ever more funds indiscriminately into R&D would not necessarily result in a profitable transformation of the Norwegian economy unless a number of other conditions also are met.

After several further rejections, the Parliament at long last voted for the establishment of the Norwegian Institute of Technology (NTH) in 1910, in the city of Trondheim, after a considerable battle that was typical for Norwegian research and science policy: the location of institutions. The choice was between the hub of Norway, the Oslo-area in which the majority of people live and work, and a smaller locale with high aspirations. The Oslo representatives in the Parliament have nearly always been outvoted in such issues, although there have been a few exceptions to this general trend.

The establishment of NTH was a considerable success. The Parliament's hesitation before NTH was approved reflected a lack of belief in the merits of technological push. This seems to be the prevailing attitude today as well, outside the circles of scientists. Thus, the establishment of NTH was not an expression of much more than an acknowledged need for engineers.

Norway never became industrialized in the same way as, for example, Sweden. Maybe that has been a blessing in disguise. Swedish industry is to a large degree locked into areas of production with ever increasing competition from the new economies, making it necessary to run faster on the treadmill every year, without being able to look forward to increased profits. It is, however, tempting to speculate whether Norway's limited industrialization in any way was due to the late establishment of relevant higher education and research institutions. This type of question is, as mentioned above, also relevant today: Should Norwegian R&D activity be higher than it is at the present time to pave the way for future knowledge-intensive businesses? To what degree is it possible to change the course plotted by historical developments, and how risky would such an attempt be?

A strong instrumental belief in the value of science prevailed after World War II

The first half of the 20th century was a difficult time for Norwegian science and research. Norway was still one of Europe's poorer countries, and the public purse was very small. The Great Depression of the 1930s, which actually did not affect the whole of Norway in a severe manner, nevertheless was a difficult period for Norwegian R&D, and this was followed by a financial standstill during World War II. In addition, there was not much belief that public investment in science would have any substantial economic impact. Despite this, several new research institutes were established during this period, mostly as cooperative initiatives to serve particular industrial sectors, such as, for example, the Paper and Fiber Research institute in 1923, and the canning industry's Research Institute, established in 1931. In the public domain, the Norwegian Polar Institute, with roots that go back to 1928, must be mentioned as well.

The development of Norwegian science and R&D erupted after 1945. The tone was set in 1946 by Prime Minister Einar Gerhardsen who stated: “The industrial competition between nations has begun to be a race in technological and scientific research. We must now vigorously recover lost ground”.

It should be noted that the prime minister himself had completed no further education beyond that of elementary school. He was however voicing his own and the Labour Party’s instrumental belief in science and technology. Actually, the Labour Party was the first to mention science in its party programme, and this as early as 1933. In one paragraph, the programme voiced the need for “public contributions to the livings cost of pupils outside home, and the protection and support of research”. The happy times for Norwegian R&D would stretch until the beginning of the 1960s, with a strong belief in both technological and social engineering.

The year 1946 was maybe the most eventful year ever for the Norwegian R&D system. The University of Bergen (UiB) was established, NTH’s budget was doubled, the Norwegian Defence Research Establishment was set up, as was the Institute for Nuclear Research (now the Institute for Energy Technology).

Norway’s first research council, the Norwegian Research Council for Technology and Natural Science was also established in 1946. The Norwegian General Scientific Research Council and the Agricultural Research Council were both set up in 1949. The research councils received a substantial amount of their budgets from the profit generated by the state-owned gaming company Norsk Tipping AS that was founded in 1948. In certain periods, the level of public R&D-funding seemed to be more dependent on the population’s gambling fever than on any political understanding of the need to invest in science. It is slightly amusing to note that sin, as when UiO was established, actually also was an issue when Norsk Tipping was set up. The discussion in the Parliament was not how much of the profit should be reserved for R&D, but whether it was morally defensible to tempt the population, especially youth, with gaming activities through a state-owned company.

The dedication, speed, and achievements in these early post-war years were very impressive. Whoever would have thought that a small nation such as Norway could build and run a nuclear reactor, and thus position itself for the benefits of the nuclear age that so many people thought was part of a bright future?

At the time it was difficult to believe that a nation that was virtually demilitarized before the war could develop a high-technology weapons industry, beneficial both for civil purposes and for exports to allies.

Indeed, the events between 1945 and 1950 really set the course for Norwegian science and R&D for many years. Industrial development was closely linked to these early efforts, as was also the development of the modern welfare state, which to a large degree was built on research-based knowledge.

One contributing factor to the post-war success was probably that the R&D sector was governed by a tight network of political comrades and acquaintances. This ensured good coordination and efficient and rapid decision making, well-linked into the political establishment. The modus operandi of the post-war heroes in the field of R&D policy would be regarded as unacceptable according to today’s nearly religious norms for good governance. The results were however impressive. And it is worth noting that Finland's successful R&D policy over the last 25 years can, to a large degree, be linked to the network of leaders coordinated by the prime minister.

The proliferation of research institutions

Norwegian politics has for many years partly been founded on the axiom that it is beneficial for the nation that people live in all parts of the country and not only clustered around the three largest cities. There also is an understanding that educating youth in the regions tends to keep them there, and that research institutions are important for business and the public sector. This, more than an academic affinity, has led to a considerable decentralized growth in the Norwegian knowledge sector after World War II.

Oslo already had its university, and NTH in Trondheim was at university level. Bergen's turn arrived, when the University of Bergen (UiB) was established in 1949. At the time, the expectation was the establishment of an outward-looking institution beneficial to the region, an expectation that UiB has lived up to. The first proposals for a university in Bergen were presented as early as the 19th century. At the time, two main arguments were used. One was that a new university was necessary to stem the brain-draining effect the UiO had. The second argument was that a university was just as important for the economical development of a town as good communications with the rest of the country. Over the years since then, there has been a continued expansion of universities, colleges, and research institutes in the regions, much of it based on the belief that such institutions are important for the settlement pattern that is sought.

The University of Tromsø (UiT) was established in 1972. This university was not created because the nation needed yet another university for scientific purposes. To the contrary, it contributed to academic fragmentation in Norway. The arguments put forward by the intellectual elite against a new university were much the same as the king's concerns in relation to creating a university in Norway in the 19th century. Politicians, however, regarded a university in the northern part of Norway as a necessity for the further development of the region and also for ensuring good healthcare there. It must be added that the UiT, although the smallest of the four broad universities, has succeeded remarkably well, both in relation to the regional objectives and not least of all academically in some fields.

Maybe the most important development of the higher educational system in Norway in modern times has been the creation of numerous colleges since the 1970s, spread throughout the country. This has in part also led to the establishment of research institutions that, in the beginning, were closely associated with the colleges. In all, there are now 6 universities, 26 state colleges (after the merging of 98 colleges in 1994), and some 110 research institutes. If each institution were given a red dot and put on the map of Norway, the emerging picture would resemble a tie with randomly placed polka-dots. Although many might believe that this development is typical for Norwegian R&D policy, governed by a combination of a lack of plans and a certain feebleness of implementing whichever plans have been adopted, one probably would also find the same random growth pattern in many other European countries.

Whether planned or not, the development has been dramatic. The number of students at colleges and universities has quadrupled over the last 40 years, and the research effort has also more or less quadrupled. The research institute sector that serves public authorities and businesses was nearly as large as the university sector, measured in terms of R&D (disregarding teaching) in the 1970s, but has since lagged slightly behind, due to the continued increase in students who also are decisive for the research capacity at universities.

The creation of a number of new regional colleges in the 1970s, supplementing existing teachers' colleges and engineering colleges, some of high repute, had a considerable impact on the education level in Norway. In actual numbers, the colleges have more students than the universities do. When the colleges were established, they were supposed to be different from the universities, providing shorter educations more directly suited for work life. Academic drift was stated to be out of the question. However, the setting up of the regional colleges was a letting loose of forces that Oslo could not tame. Research institutes popped up around the colleges. And a number of colleges wanted to move in an academic direction, by doing research, employing professors, and awarding PhD degrees. Several of the colleges had university aspirations, and two of them were actually granted that status–Stavanger in 2005 and Agder in 2007. For the time being, the government has stated that it will not grant university status to more colleges. It remains to be seen how long this decision will last.

The development thus described has undoubtedly contributed to a fragmentation of Norwegian R&D. On the other hand, from a regional policy point of view, the establishment of colleges and research institutes has been a success. The same can be said in general about the education offered, although recent evaluations have concluded that teacher-training and nurse-education in part is substandard, a fact which is now of national concern. The development also illustrates the lack of willingness or ability demonstrated by central authorities, to steer the course of events in accordance with stated policies. The flip side of this coin is flexibility, pragmatism, and finding solutions that fit in locally.

Norwegian R&D–International orientation

It would be unfair to say that Norwegian R&D policy only has had a regional perspective. In reality, Norwegian scientists have always had an international orientation and have been integrated in the European scientific web since the 19th century. Darwin's Origin of the Species was, for example, widely read in Norwegian academic circles the year after it was first published and was used by some Norwegian natural scientists as the ultimate argument that the university should turn away from Christianity as the basis for its existence.

After World War II, most scientists who wanted to be taken seriously visited scientific institutions in the USA, for times at great length, often leaving their children behind slightly neglected in the care of nannies or maids. The United States generously funded visiting scientists, and Norwegian authorities chipped in as best they could. The links between Norwegian scientists and their colleagues in the USA are still much closer than what appears to be the case when looking into the formal state-to-state R&D agreements. A contributing factor might be an advantageous tax agreement for scientists between Norway and the USA.

The international orientation of Norwegian R&D is also expressed by the formal membership in different international R&D organizations and research facilities. Norway was thus one of the founding members of European Organization for Nuclear Research (CERN) in 1954, became a member of European Molecular Biology Laboratory (EMBL) in 1986, a member of European Space Agency (ESA) in 1987, and has participated in the European Union's (EU) Framework programmes for RDI since 1994. Historically, CERN has not only been important for Norwegian basic science research, but also has opened the door for the Norwegian high-tech industry. The participation in ESA has led to a development of the Norwegian space industry, which is much more important in scientific and industrial terms than most people are aware of.

The participation in the EU Framework programmes has been a success in two dimensions. The first one is that Norwegian R&D is becoming ever more integrated in European networks, stimulating quality and the transfer of knowledge. The second dimension is that the return-rate more or less is the same as the Norwegian financial contribution to the programmes, indicating that Norwegian R&D is competitive. The publication of scientific papers in collaboration with scientists abroad is furthermore increasing year by year.

The participation in the EU's Framework programmes has influenced Norwegian R&D policy in many ways. In the long run, maybe the most important effect lies in the fact that the participation is a constant reminder that science is an international commons. At a policy level, priorities expressed in the Framework programmes have influenced Norwegian priorities, and the instruments used in deploying R&D policy have in part been influenced by the running of the Framework programmes.

The acceptance of quality as a decisive factor in R&D policies

Over the years, the participation in the EU Framework programmes has had a considerable mental impact on Norwegian R&D and R&D policy. They have made it ever more evident that Norwegian R&D policy is part of and dependent on what is going in the rest of Europe, scientifically speaking. It is also clear that it is not sufficient to be a world champion in Norway if you cannot compete abroad. The increasingly close integration in European R&D has made the field of scientific competition more visible than before.

In an egalitarian country, one does not focus much on excellence, except in sports and music. Although everybody has known that some scientists belong to an elite group and that most are mediocre, there has traditionally been little talk of this in

Norway, and all scientists have in principle had much the same conditions for their research.

In part due to international influence, but probably also, in part, a reflection of a more general cultural change, it has now become quite legitimate to talk of excellence in R&D policy, and to do something about it if it does not exist. When the idea to establish Norwegian Centres of Excellence was presented some ten years ago, there was concern that the concept might be unfavourably received. But contrary to what some feared, the idea was embraced by the scientific community. The whole process of selecting and setting up the first Centres of Excellence some nine years ago was an undisputed success, and since then, the centres have done impressive work. It came as rather a surprise that there actually were more groups of scientists in the international top league than expected, and that there was scope for increasing the number of Centres without compromising scientific quality.

Generous funding for the best was the main aim of the new scheme, but not the only one. The host universities had to chip in to finance the centres. They were thus induced to prioritize the elite to a degree that they hardly would have managed on their own accord.

The growth of research institutes

A defining characteristic of the Norwegian R&D system is, as already mentioned, the large presence of independent research institutes. The creation of the institute sector was not the result of any master plan, but something that just happened. The sector is extremely heterogeneous, and thus difficult to classify. A number of institutes (at present 65), with different formal statuses, receive core funding from RCN, and are thus regarded as part of the public research and innovation system.

Dominant players in the field are termed the technical-industrial institute sector. The two largest institutions are SINTEF and the Institute for Energy Technology. Indeed, in size, SINTEF is a considerable European research institute. The creation of these two institutes goes back to the first years after World War II.

The Nuclear Research Institute was established first as a spin-off from the Defence Research Establishment. As nuclear energy gradually fell out of favour, the institute changed its name to the Institute for Energy Technology and developed competence in a number of new fields, thus adding many new arrows to its quiver of knowledge. The institute still runs its reactor, but now as an international programme focusing on reactor safety and process control.

In the late 1940s, both industrialists and politicians came to the conclusion that there was a need for a central industrial research institute (SI). This institute should not be dedicated to any particular technology or branch of industry, but should provide research and consultancy services within a wide range of topics. The proposal was to establish such an institute in Oslo. Naturally, this choice was heavily contested by NTH in Trondheim, as NTH was of the opinion that the only location that made sense was Trondheim, close to the highest technical education institution in the country. When SI eventually was established in Oslo in 1949, NTH was more than irritated, and created its own research institute, SINTEF, in 1950. SINTEF lived in close symbiosis with NTH for a number of years before it gradually gained independence from its mother, although the links are still close. Over the years, SINTEF merged with a number of smaller research institutes, crowning this development by taking over SI. Thus, Trondheim lost the battle in 1949 but won the war in 1993.

A number of institutes were established to perform R&D of particular interest for sector authorities. As examples, one can mention the Norwegian Institute for Water Research, the Norwegian Institute for Air Research, the Norwegian Institute for Nature Research, Nofima (covering important aspects of fish research and food research), the National Veterinary Institute, the Institute for Social Research, the Work Research Institute, and the Institute of Transport Economics. In this context, it must be mentioned that many institutes which originally were perceived to be regional because they were located outside the larger cities, are actually also to some degree national institutes, providing services to partners all over the country. One example of this is Møre Research, located in Molde in the county of Møre and Romsdal. Several of these institutes still provide valuable services to local authorities in their vicinity.

Pundits of R&D policies have occasionally claimed that the institute sector is much too large. According to this thinking, the existence of the institutes is a disincentive for businesses to do their own R&D and thus develop their competence. On the other hand, the institutes supposedly shield universities from society's challenges, allowing the professoriate to remain inside the ivory towers and ascend to ever higher hemispheric layers of science. From a historic perspective, there is, however, little merit to these lines of thinking.

Most Norwegian businesses have been so small that it very often made much more sense to call on the assistance of a research institute rather than trying to do the job themselves. There have always been some technologically advanced businesses, and they have had the choice of relying on either their own efforts or the help from a research institute. The biggest companies have to a large degree done their own R&D.

The presumed isolation of the universities is a slightly more complicated issue. First of all, quite a lot of scientific research, however important it may be in the long run, is not of direct interest for businesses and public authorities, and the scientists might quite legitimately not have the aptitude for consultancy or applied research. Furthermore, the core business of university scientists is teaching and basic research, and many do not have the time to do much more. (It is, however, interesting to note that in a number of fields, the best teachers and scientists also are heavily engaged in extramural scientific activities, beneficial for both science and society). This being acknowledged, individual researchers in a number of disciplines of practical interest have been more heavily engaged in applied research and consultancy than what appears to be the situation when reading universities' annual reports. Thus, cooperation between university scientists and research institutes has been the normal course of affairs, both on an ad hoc basis and through part-time positions at the university or vice versa. A good example of this is the relationship between the Simula Research

Laboratory and UiO, an expression of which is the close collaboration seen in the frequent joint publication of scientific papers.

Simula Research Laboratory is a new type of research institute, somewhere in between the traditional research institutes and the universities. Simula has a high level of government funding and no formal obligations to generate income or teach students (although both are done), and has been given renewable five year contracts on condition of a favourable outcome of an international evaluation. The Sars Centre in Bergen is probably the only other comparable research institute in Norway. The evaluators of the Sars Centre have lauded it for its scientific achievements and have regarded the unbureaucratic management structure as a blessing. Both institutes have restored the scientists as full time investigators.

The establishment of these two institutes, which indeed are governed in a differing way than are the universities and that enjoy better economic framework conditions than other research institutes, simply *happened* without any deep policy discussion beforehand. This popping up of new types of institutions can either be regarded as the unlucky consequence of slack government control in the field of R&D policy, or as a happy example of how government accepts variation without much fuss and lets evolution take its course.

Historically, the relationship between the universities and the research institutes has been a rather harmonious one, with a rather low degree of competition for funds and glory, beneficial for both institutions and for businesses and public authorities. The question is whether this will last, or if the research siblings will become destructive adversaries. On the one hand, universities are now expected to be more active in technology transfer, or put in another way, secure intellectual property rights more effectively than they have in times past, and also, exploit them. On the other hand, the funding system for the institutes is being modified, putting more emphasis on academic achievements. It is well-known that different species can coexist peacefully if they occupy different ecological niches, but that vicious behaviour erupts if they have to compete for the same resources. History will show whether the past's coexistence will continue, or whether an ecological struggle with an uncertain outcome will ignite in the years ahead.

The establishment of Simula Research Laboratory

The history of the way in which the Simula Research Laboratory came about is a good example of how circumstances without a plan dictate events. Around the time that the old airport at Fornebu close to Oslo's city centre was going to be shut down, a large and attractive piece of land by the fjord was ready for new uses. A well-known ship owner launched the vision of IT Fornebu. The ambition was to develop the area into an international centre for future information technology, attracting the best of businesses and researchers from all over the world. The idea was not only to set up office buildings, but also to integrate very attractive dwellings bordering the beautiful shoreline. The proposal was that the Norwegian government and private industry should create IT Fornebu as a joint undertaking.

The Labour Party, always a firm believer in government's ability to direct economic development, seized the opportunity to join forces with industry to create something new and ambitious. The question of government support became quite a political issue, and in the end the Labour Party secured a majority in the Parliament with the help of the populist and conservative Progress Party, against the wishes of all other political parties. Part of the project was to establish a new research centre for information sciences at Fornebu.

The Research Council of Norway was in favour of creating the new institute, but hesitant to become publicly involved in anything that was related to the hot political issue IT Fornebu. With a minority government, it is always uncertain who will be the masters of the research purse next year, so it was best not to get involved in political brawls. In addition, the Research Council was, at the time, heavily involved in developing its own properties close to the University of Oslo. A public engagement in the disputed plans for IT Fornebu could highlight the possibility of stopping this development and relocate funds to the further development at Fornebu.

The four broad universities and major research institutes, always opportunistic when it comes to securing new funds, got their act together, took the lead, and came up with a proposal for the new research centre. The proposal suggested the institute's scientific content and how it should be financed. According to the plans, the universities and relevant research institutes would own Simula Research Laboratory together.

What happened was that the vision for IT Fornebu burst with the dot.com bubble. But Simula Research Laboratory was established and given generous government funding. Simula was, for many years, the only element left of the original proud plans for creating something new at Fornebu, intended to push Norway forward as a leading IT-nation. The symbolic expression of the collapse was the nearly empty terminal building at Fornebu in which Simula, for several years, was the primary tenant.

On passing the proposal for establishing the Simula Research Laboratory over to the Parliament, the Ministry altered the rules of ownership. The universities were shoved out, and the Ministry decided to take ownership of 80 per cent of the shares itself, the remainder being owned by SINTEF and Norwegian Computing Center. The reason for this change of ownership, given informally at a later stage, was that the state (in this case represented by the universities) could not be in disagreement with itself at shareholder's meetings of Simula, and therefore the Ministry had to hold all the shares owned by the state. There is little legal merit to this line of thinking, but the result makes sense. In general, it is not wise to have competitors as large owners of a company, and the same applies for research institutions.

The interesting question is whether Simula will turn out to be a non-viable mutation amongst research institutions, or the first specimen of a new species that will contribute to Norway's scientific success in the 21st century. Is it sensible to have research institutes, which can devote themselves to science without the teaching burdens of universities and without the need to generate income in the market place, be governed more like a private company than a public body?

R&D policies dictated by educational needs

As mentioned above, universities have more or less quadrupled in size during the last 40 years, as measured by the number of students, employees, and budgets. This means that basic research has grown proportionally as well. The allocations to universities are based on the presumption that half of the scientific personnel's time should be dedicated to research. In principle, this should afford a rather unique opportunity to prioritize (at any rate to a certain degree) basic research in a direction that is thought to be beneficial for society in a long-term perspective.

Alas, this has not been the case. The university growth has not been dictated by scientific needs, but rather been a consequence of educational politics. The prevailing political view has been that young people, as far as possible, should get the education they want, and that universities should expand to meet the youthful demand for education. At the same time, university policy and the size of the budget has been based on the principle that university teachers should allocate half their time for research. The time for research has historically not been regarded as an institutional asset, but a personnel right for the scientist. Combining these facts, it is therefore correct to say that youth just out of school to a large degree have directed the growth in the different disciplines, and that research takes place wherever academic education is offered.

The result of the described policy is undoubtedly a considerable fragmentation of research in Norway into many small groups that do not get above the critical mass necessary to excel in science. In R&D policy terms, the period 1945–65 could be termed the happy years of expansion and scientific optimism, and the period 1965–90 could be called the period of student-directed growth in science, uncoupled from other national needs.

This picture is not absolutely true. The universities have historically skewed their appropriations slightly compared to the student influx to different studies. But there have been two limiting factors. The one is the mantra that university education should be research-based, interpreted to mean that the teachers should be active researchers. The other factor is that organizations based on self-governance usually have great difficulties in prioritizing at the cost of colleagues and even greater problems in deploying any such decisions that might have been made in brief moments of courage. Taking this into consideration, a relevant question is how long can the historical tradition of democratic university self-governance continue. No other important national organizations are governed by the employees and inmates.

In recent years, the universities' framework conditions have been altered. The government funding system is now more transparent and to a higher degree decided by results achieved. This seems to have led to a clear increase in international publication, raising Norwegian science considerably in the ranking list. Universities now also compete to attract as many students as possible. This might be a good thing. However, using ever more money on advertising might also turn out to be a costly type of academic zero-sum game. The principle of using half the work time on research is no longer an individual right for university teachers. It is too early to say whether universities will use the freedom to allocate research time strategically. This is a field of tension between university needs and the interests of the individual scientist. At present, the universities do not seem to be facing the new music they can play, but continue drumming on a lot of scientific deadwood. This is in part due to the system of self-governance.

The struggle between research councils, ministries, and the scientific community

Who, in addition to the students, directs the science priorities in Norway? The question has no easy answer.

At the highest level, of course, government and the Parliament make the decisions. But this final wisdom is clearly very heavily influenced by the different ministries, due to the sector principle. The sector principle has been explicit the majority of the time since World War II. According to the principle, each and every ministry is responsible for the organization and funding of R&D within its sector. Thus the Ministry of Oil and Energy is responsible for ensuring that the right level and type of R&D takes place to increase the extraction rate of petroleum in the future, the Ministry of Fisheries and Coastal Affairs for fishery-relevant R&D, and the Ministry for Research and Education for basic research, and so on.

The sector principle has historically lead to virtually all ministries getting involved in R&D matters, which indeed is positive and contributes to keeping R&D relevant for the sector. However, because Norwegian ministries in practice are rather independent of each other (except the Ministry of Finance, which is regarded as a threat to all, as in most countries), the tendency is to have the same number of research policies as number of ministries. In principle, this is not a serious problem as long as some part of government ensures coherence. All ministries intellectually support the idea of national coordination and priority setting in R&D policy as long as nobody interferes with what they are doing. The Ministry of Research and Education is entrusted with the task of coordinating government R&D policy. But alas, the Ministry has no instrument of power other than persuasion, which indeed often is insufficient in meetings with headstrong ministers used to running their own show, unless the research minister in power is a figure with considerable general political influence.

Most governments have, for the last 40 years, established a Government's Research Policy Committee, consisting of the ministers who are most engaged in R&D-matters. This could have been an important place for drawing up national policy, for setting priorities, and ensuring coordination. However, for a number of reasons, it seems correct to say that this potential has rarely materialized, although there are some important exceptions to this. One of the reasons is that no single R&D budget is proposed by government. Instead each and every ministry has to fight for its R&D money within the limits set for the ministry's total budget. Appropriations for R&D therefore have to compete with other and often more popular ways of spending the ministry's money. The Government's Research Policy Committee is not the right place for discussing the different ministries' total budget profile. In addition, no long-term national budget for R&D has ever been proposed. Such a budget could possibly induce ministries to become more engaged in developing a holistic national R&D policy.

With the growth of public spending on R&D, most countries have established research councils, although the names and the tasks vary slightly from one country to another. The core business of Norwegian research councils has been to process applications for research funding and to allocate funds on a competitive basis. The research councils have also always argued for more government money for R&D. In addition they have, at times, attempted to offer advice to government in R&D-policy matters.

Over the years, four research councils were established in Norway, and in addition two other organizations with research council tasks on a more limited scale. In 1993, they were all merged into the Research Council of Norway. The RCN has responsibilities in all fields of science, both for basic and applied research, and for business R&D. The RCN is a singular European institution with its very wide mandate, as most countries have several different research councils.

The life of the previous research councils was not an easy one. In part, they were competing amongst each other for funds and for influence on government R&D policy. Furthermore, the research councils wanted to receive funds with as few strings attached as possible, whereas ministries would not let loose their control. Thus, there was a continuous tug-of-war between the research councils themselves, and between them and the ministries.

There was also tension between the research councils and the scientific and business communities. The beneficiaries at the receiving end were suspicions of favouritism, sceptical of the fairness of procedures and, not least of all, of the priorities made by the research councils. There was also often an uncomfortable relationship between the administrative staff and all the external board members or committee members, who made the formal decisions of the research councils. The external members felt they had a legitimate right to run the show. The fixed staff would often regard the external board members as guests pursuing their own interests, and that a firm hand was necessary to ensure due process, run a tidy ship, and attempt to be consistent over time. One of government's frustrations was that it supposedly wanted holistic advice from the research councils, but instead received as many opinions as there were councils. The research councils did, at times, attempt to coordinate themselves in policy matters, but with marginal success.

After a tumultuous start, the RCN has been chugging along at a steady pace, and in recent years the situation has been relatively calm in relation to all the stakeholders. However, most of the problems the old research councils grappled with still persist, although generally to a lesser degree than before. The quarrels between the old research councils are now a question of how to manage internal processes, balancing alternative perspectives and interest within a single research council. The persistent problems are not due to the organizational set up of RCN, but are inherent to the task of a research council, the way the ministries function and earmark budgets, and the political and administrative culture in Norway. As to the latter, it does not help much for the RCN to send a proposal for a holistic R&D-policy to government if there is no one interested at the receiving end.

Nudging forward in the right direction

Norwegian R&D policy has never been based on precise plans or clear visions. Quite often developments have been the opposite of what was the stated policy. And big words have sometimes remained just words, as, for example, the objective of increasing Norwegian R&D up to three per cent of the GDP, a goal with which all political parties have agreed. The R&D system is fragmented with many small-sized research groups. There is a continuous struggle and competition for funds, glory, and influence. The management of universities is not impressive, and the coordination of sector R&D policies is weak.

On the other hand, Norway has created a R&D system and policies with considerable variety. The R&D policy is based more on an evolutionary process than intellectual finesse. Evolution is not beautiful, however attractive the end results otherwise may be. Yet the Norwegian R&D system is flexible, adoptive, and reasonably well-tailored to the needs of the nation. Norwegian researchers do well in the EU Framework programmes for RDI. The process, when choosing the centres of excellence, in 2003, which was based on international evaluations of applications, demonstrated that there were absolutely more first-class researchers in Norway than what had been previously believed. Scientific publications and international citations have increased substantially over the past several years.

So, all in all, it would seem fair to say that Norwegian R&D policy has persisted in moving ahead, albeit in a slightly haphazard way but with a number of successes to its credit. And although stringent plans have perhaps been lacking, and even if development has been uneven and difficult to describe in a comprehensive way, it is possible to identify different epoch in modern Norwegian R&D-policy. Thus the period 1945–65 could, as already mentioned, be coined the happy years of expansion and scientific optimism, and the period from 1965 to 1990 could be called the area of student-directed scientific growth. From a historical perspective, it is premature to characterize the recent past and the present. But in the future, the period from 1995 to 2015 may well be described as the era of quality and internationalization, and it is hoped, not the time of lost opportunities.

Whether Norway could have achieved success better with any other R&D policy than the one that has reigned, is an open question. The really interesting issue is which development of Norwegian R&D policy is needed for the future. Alas, this question falls outside the scope of the present text, however much this author is tempted to air his opinions.

11

SIMULA — THE LANGUAGE

Olav Lysne and Are Magnus Bruaset

Naming a Research Laboratory

Walking the corridors of the Department of Informatics at the University of Oslo in the mid-1980s, one could see large stickers on several office doors claiming that “Simula does it with class”. This bold statement was a play on words that literally pointed to the fact that the programming language Simula embodied the important paradigm of object-oriented programming through defining the concept of classes. Metaphorically, the statement demonstrated the department’s pride of knowing that a quantum leap in the theory and practice of programming languages was due to two persons in its midst, professors Ole-Johan Dahl and Kristen Nygaard. The language had an elegance that promoted code aesthetics and their research contribution was undeniably world class.

Simula Research Laboratory is named after the programming language Simula. Therefore, it is quite logical that when we present the laboratory, we are often asked whether we still do research on the language. The answer to this question is no. As a programming language, Simula is hardly in use anymore. However, the most important new concepts that the language brought along–objects and classes–are cornerstones of modern information technology and the world would arguably look quite different without it.

The reasons why a new research laboratory at its birth in 2001 took the name of an old programming language were twofold. First, it was intended to honour the outstanding scientific achievement of Dahl and Nygaard[1]. The second reason was to

Olav Lysne · Are Magnus Bruaset
Simula Research Laboratory

[1] In 2001, Dahl and Nygaard were jointly awarded the IEEE John von Neumann Medal by the Institute of Electrical and Electronic Engineers “for the introduction of the concepts underlying

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_11,

demonstrate the new laboratory's ambition to perform important research that met high standards of quality.

This chapter gives some background on the programming language Simula. It briefly presents the concept of object-orientation and how it was conceived. Finally, it discusses the heritage of the work done by two brilliant Norwegian scientists in the 1960s. For a more detailed discussion of the development of the Simula language and the remarkable events and processes that surrounded its conception, we refer the reader to the historical exposition by Jan Rune Holmevik[2] and the references therein.

The Concept of Object-Orientation

When the Simula language was developed, the main limiting factor of software development was the ability to handle complexity. Several generations of programming languages had been devised, each trying to bridge the gap between the sets of instructions computable by a machine and concepts tractable to the human mind. Then as now, each machine came with an *instruction set*, each instruction describing a basic operation such as adding two numbers or storing a unit of data in a particular memory location. Each computer program thus consisted of a sequence of such instructions. It is the sum of instructions that makes the computer do something useful. In the beginning, each instruction was identified by a sequence of *bits*, each bit denoting a binary state–a 0 or a 1. Remembering the semantics of binary instructions, however, turned out to be difficult for most programmers. Therefore, understandable names were given to basic machine instructions such as *add* and *store*. Now, programmers could write programs using these names, or mnemonics, instead of the actual instructions and have a separate program translate their code into the binary representation the computer required[3].

The third generation of programming languages attacked two other sources of complexity: the abstraction of physical memory locations and data types through the concept of *variables*, and the control of program flow. In all areas of programming there are portions of code that need to be executed several times, either in succession or at specific points in the computations. Instruction sets for a particular family of computers would control program flow through instructions such as *jump* or *goto*, which simply moved the point of execution from one place in the code to another. Although goto statements still persist in modern programming languages, in practice

object-oriented programming through the design and implementation of SIMULA 67". Shortly after, they received the 2001 A. M. Turing Award from the Association for Computing Machinery "for ideas fundamental to the emergence of object oriented programming, through their design of the programming languages Simula I and Simula 67."

[2] Jan Rune Holmevik, *Inside Innovation–The Simula Research Laboratory and the History of the Simula Programming Language*, 2004. See simula.no/about/history/inside_innovation.pdf.

[3] These languages go under the name of assembly languages and still exist today. They are used when there is a particular need for computing speed that cannot be supported by a high-level language.

they have been almost entirely replaced by third-generation notions of *while* loops, *for* loops, and *methods*.

It was into this universe that the concept of *object-orientation* was born. In essence, most, if not all, computer programming consists of describing some aspect of the real world in a way that allows the computer to mimic it. From this observation one can deduce that useful concepts for describing programs would be those that are well suited to capture the essence of real-world objects. In the Simula language, this was done by a construct that encapsulated the *state* of an object through a set of variables, its *relation* to other objects through a set of pointers, and their *actions* and *interactions* through methods that were considered to be internal to the objects themselves.

The power of this concept in programming is hard to convey through mere written text. It did, however, lead to a whole new way of thinking. Where earlier there was a separation between program data and the instructions manipulating that data, suddenly, it became natural to view the data structure as a collection of objects that were created, that acted upon impulses from other objects, and that ultimately died. For example, in a graphical program for interior decoration one could ask the "table" object to draw itself on the screen. Later, if the concept of "chair" should be added to the program, that would be an object with its own description of how to draw itself. In this way new types of objects could be added without having to change any UpdateScreen methods in the code.

The most important notions in object-oriented programming, such as classes, objects, pointers, methods, messages, and inheritance, were already defined by the Simula language in 1967. Some of the notions had other names–for instance, what we call methods today was called procedures in Simula–but the concepts were the same and have been stable since.

From Operations Research to Objects

Operations research is a field that applies mathematical modelling, statistics, and algorithms to search for near-optimal solutions to complex problems. Its first celebrated results stem from the 19th century, when Charles Babbage analysed the cost of transporting and sorting mail, but its more modern development started with the successful application of operations analysis to military operations during World War II. Research on operations analysis continued at the Norwegian Defence Research Establishment after the war and it was in this institution where Kristen Nygaard and Ole-Johan Dahl met.

Faced with the problem of using computers and third-generation programming languages to simulate operations, the two Norwegian researchers conceived the idea of defining a new language specially tailored to this task. Following the naming convention from Algol (ALGOrithmic Language), they called their new language Simula (SIMUlation LAnguage). Their first approach to this special-purpose language was to introduce a concept of

> discrete units demanding service at discrete service elements, and entering and leaving the elements at definite moments of time. Examples of such systems

> are ticket computer systems, production lines, production in development programs, neuron systems, and concurrent processing of programs on computers.[4]

This concept was the fundamental idea behind the first version of Simula, which appeared in the early 1960s.

Object-orientation as we know it today is very far away from the description above. Still, there exists a continuous path of new insights and developments from these ideas in 1962 up to the Simula Common Base Language of 1967. These insights include the observation that the roles of passive discrete units and active service elements are interchangeable, leading to a unified notion of "unit"–the *object.* Furthermore, they include the development of prefixing and inheritance from the notion of *classes.*

Most, if not all, of the concepts that we now consider to be at the core of object-oriented programming were already in existence in 1967. This is quite remarkable, since object-oriented programming was a very active field of research for decades, and is a manifestation of the quality and thoroughness of the pioneering work of Dahl and Nygaard.

The Responsibility of Heritage

Although the main building blocks of modern object-oriented programming were already defined in 1967, it took close to two decades for these ideas to penetrate mainstream industrial programming. There are several anecdotes from these early days of the pioneers of this new programming paradigm. One of the most famous is that of Alan Kay, who studied the code of what he thought was an Algol compiler. He noticed that the storage allocator was different from what he had expected. Through closer study, he learned the basic concepts of object-orientation and, from that point, went on to create the programming language Smalltalk. What he had actually gotten a hold of was not an Algol compiler at all: It was a Simula compiler.

Kay's Smalltalk was an important language on many university campuses, particularly in the United States. It was, however, another language that paved the way for the industrial use of object-oriented programming. In the early 1980s, Bjarne Stroustrup at Bell Labs enhanced the widely used C language with Simula-like features. His new language was originally named "C with classes", but it was renamed C++ in 1983. Leveraging the user base that C already had at the time, particularly on UNIX platforms, C++ was the language that gave objects and classes acceptance in the computer industry. Later this trend further proliferated through the Java programming language in use from the mid-1990s. Java is now the training programming language of choice for most universities around the world. The trend also continues with the widely used scripting language Python, which supports object-orientation alongside other programming paradigms. A curriculum in computer science that does not include object-oriented programming in some form is inconceivable today.

From the perspective of a research institution, the history of the programming language developed by Dahl and Nygaard is both inspiring and thought provoking.

[4] This quote is from the introduction to the paper 'Simula–An Extension of ALGOL to the Description of Discrete-Event Networks', published by Kristen Nygaard in 1962.

It consists of two dedicated researchers attacking the important problem of using the new and powerful invention of electronic computers to their full potential. It tells the story of how they continued working on this problem for a long time, from the first sketchy ideas in 1961 until the definition of Simula Common Base in 1967. It is also the story of thoroughness and high scientific quality. In spite of all the research that was done on object-orientation later on, very few concepts have survived that were not already thought of by Dahl and Nygaard. Finally, their work had the necessary element of practicality, in that they not only defined the language but also solved the problem of having working compilers built for multiple platforms.

The marketing of the language was, however, not such a success. Today, Simula as a programming language is history. The concepts that it gave birth to, however, have showed their value by penetrating almost all areas of computer science. By reusing the name Simula for our research institution, we recognise and honour the heritage of the brilliant piece of research conducted by Dahl and Nygaard in the 1960s. To this end, we are also determined that for the years to come, researchers should agree that "Simula does it with class".

PART II

BASIC RESEARCH

Olav Lysne, Director of Basic Research

12

INTRODUCTION TO BASIC RESEARCH

Olav Lysne

The operational unit Basic Research was created in Simula's reorganisation in 2008, alongside Research Applications and Research Education. Basic Research is divided into the research departments Networks and Distributed Systems, Scientific Computing, and Software Engineering. These three departments constitute the core of the activity at Simula Research Laboratory. The PhD students and postdoctorate fellows employed in Research Education are supervised and carry out their daily work in Basic Research, where they contribute substantially to the scientific work. Likewise, the activities in Research Applications are initiated by and collaborated upon with researchers from Basic Research.

Basic Research's aim is to conduct long-term basic research with a clear view to the future application of the results. This apparently paradoxical statement needs some explanation. Projects at Simula work on fundamental, complex problems that will remain a challenge for a long time. In that respect, what we do is basic research. Still, all projects at Simula should be such that the potential impact of their research results is high. They should study problems that are considered important not only by the research community but also by society at large.

Strong research results with potential impact are, however, not always enough. At Simula, we strive to maintain a culture where each project has a clear understanding of the processes that lead from new insight to its utilisation in society. Sometimes publication in high-ranking journals and conferences is in itself the best way to promote the results. In other cases we need to combine scientific publication with other measures.

Olav Lysne
Simula Research Laboratory

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_12,

One good example is a software engineering project aptly named BEST. This project involves improving cost estimates of software development projects and better handling the uncertainty related to these estimates. In spite of the very applied nature of the problem formulation, this is a basic research problem that will remain a challenge in the foreseeable future. With regards to publications, this project is known to be one of the most productive in the world in software engineering. Equally important, BEST has established a practice of offering a series of very popular courses for practitioners in the IT industry. It is through these courses that BEST promotes the impact of its research results outside of academia. See chapter 26 for a closer description of this project.

Another example is the project on interconnection networks in the Networks and Distributed Systems department. This activity started in the early 2000's and focussed on efficient and deadlock-free routing in arbitrary topologies. At the time, this was an unsolved problem with no present-day application. The research on interconnection networks therefore started by publishing results on network structures obtained by means of computer simulations. Currently, methods developed in this project are applied by the world's largest computers. The uptake of results from this project in industry happened in part as a result of the companies themselves finding and reading our scientific papers, and in part through our engaging in collaboration with industry and the development of open-source software. These activities are more closely described in chapter 14.

Finally, the Scientific Computing department has been conducting long-term research on modelling the heart's electrical activity. At the outset this was pure, basic research that aimed to understand and model the creation and flow of electric signals through the human body. As research proceeded, it spawned a project that investigates the extent to which the gained knowledge can be used to identify early stages of heart infarction from electrocardiogram recordings. See chapters 20 and 22 for more information.

The following chapters describe many more examples of research projects in various phases of development. Although they are very different from one another, all have common traits. They all started by studying fundamental and complex problems with long-term horizons and high potential impact. At the outset the projects consisted of researchers with strong potential to create excellent research results and publish these in academic outlets. Some of the projects have, after years of work, reached a point where the results are sufficiently developed for their potential impact on society to be investigated. This span is what basic research at Simula is about.

13

NETWORKS AND DISTRIBUTED SYSTEMS — WHY, WHAT, HOW AND WHAT'S NEXT

Carsten Griwodz and Olav Lysne

Why we do Research on Networks

Networks today are an essential part of the communication among people and among and within machines. They are expected to enable immediate and unconstrained communication with anyone anywhere. While the technical ability for such communication is desirable, it is neither feasible nor possible. The two constraints that we cannot overcome in our efforts to reach this goal are perceived as speed limited: communication cannot be faster than the speed of light and the communication bandwidth is limited by the available spectrum. Within these constraints, however, there is still a huge potential for improvement. There is an increased demand for networks and distributed systems on smaller scales and this demand is also due to physical constraints. Ever since the miniaturisation of transistors reached the atomic scale, performance increase is no longer easily achieved and communication within and among chips is the established approach for achieving further improvements in computing power. Networks and distributed systems investigates the many promising options for achieving such improvements at various scales of communication.

Carsten Griwodz · Olav Lysne
Simula Research Laboratory

Carsten Griwodz · Olav Lysne
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_13,

At the very small scale, networks on a chip are used to connect highly integrated, independently designed microchip components. Such chips can consist of many highly specialised components that need to be interconnected or they can be multicore chips, where each core is essentially implemented like a stand-alone but only partly specialised chip. The high level of integration leads to problems with yield and runtime errors and network research at this small scale is required to handle these failures. Beyond this issue of robustness of both production and use, the efficient use of such chips is also important. Investigating how programs can be executed on parallel processors provides better ways of partitioning them, that means assigning parts of programs to parts of the chip at the right time. This reduces computing resource waste, often decreasing the time it takes to complete a computation, but this can also allow the parallel execution of several programs or shutting down parts of a chip to reduce its energy consumption. With the rapid spread of such chips to desktop and even mobile computers, research in this area is intense and urgently needed.

Cluster computing is usually a means of reaching the highest performance for the largest applications, which require computing power that is millions of times greater than can be achieved by individual chips. Historically, few such clusters have existed, performing important jobs such as predicting the weather and performing nuclear simulations. Under several names, however, most recently "cloud computing", clusters are increasingly becoming a means for providing computing power as a commodity. Clustering is a good means of simplifying the maintenance of a large amount of computing power provided for handling a multitude of less demanding applications in parallel. The research questions that exist about improving the operations of clusters are similar to those of networks on a chip but the answers are quite different. Here, component failures are expected to occur regularly and can often be detected and fixed. Communication among the parts of a cluster must survive such failures, with the smallest impact for the applications. Better decisions about the mapping of applications to parts of a cluster can reduce their runtime and the energy thereby required.

Networking between individual computers affects general users more frequently than the issues above. Whereas wired local area networks are dominated by the Ethernet family of protocols, wireless networking is a field that is wide open for research. Regulators' recent release of the spectrum, from analogue television among other things, is apparently providing for the higher bandwidth craved by consumers, but the best means of using it is still a matter of research. First of all, bandwidth and low latency in wireless networks are generally unstable. Mobility of the communicating nodes, physical obstacles, interference, and unregulated access pose major challenges to the predictability of service. Research on improving wireless networks is therefore ongoing on all levels, ranging from the understanding of interference and ways of overcoming it to adaptive applications that are able to use several wireless networks at a time.

Connecting entire networks is an even larger issue, but the Internet has apparently been an ideal enabler for such endeavours. The Internet has, however, grown from its origins of a small number of very different computers and networks that needed to be connected in a pragmatic and affordable manner. Its simplicity and limitation to the most basic of rules and regulations, which is the reason for its success, is also its biggest weakness. While a huge research effort is currently invested in the development of a new Internet, it is very important to realise that today's Internet is used by 22 per cent of the world's population and that the inertia of such a huge number is nearly impossible to overcome. Improving the Internet therefore requires evolutionary changes and not revolutionary ones. While a revolution can start with a blank slate approach, evolution requires a deep understanding of the Internet's current operation and reasons for its limitations. This includes the ways in which routes are maintained among the growing numbers of networks, the reasons for routing changes, and the means of achieving low latency and sufficient bandwidth for distributed applications within the current constraints.

Using all of these networks in the most efficient manner cannot be considered a networking issue alone. Understanding user requirements and helping to address them in efficient ways are in many cases more appropriate tasks. Considering that the bandwidth is limited, it is certainly helpful that a distinction already exists between the 120 MBit/s of studio-quality video and the 4 MBit/s needed for the high-quality video watched by consumers. This distinction, however, is not enough for streaming such video to the increasingly heterogeneous devices that are used to connect to the Internet. Handheld devices can't display high-quality video and neither can today's wireless networks support these data rates for several users at a time. Adaptation is therefore required. Although streaming video has been the killer application of the last decade, new applications with different challenges constantly arise: For example, interactive applications require consistently low latency, while immersive reality requires data rates far beyond those of streaming video, as well as interactive elements.

Research needs to provide the means for scaling systems to handle the computing power required by these applications and for connecting them across networks. Ways of developing such applications in a scaling- and network-friendly manner need to be investigated. In a distributed environment where physical placement isn't necessarily the concern of developers or users any more, researchers must also understand the appropriate physical placement of the components used in distributed applications.

Simula's Networks and Distributed Systems department addresses many of these issues in its research. The department works on practical problems whose impact on real-world implementations is highly probable. Networks and networked services permeate our daily life and their performance and reliability are essential. From mobile telephony and banking to Internet television and music downloads for individual end users to the ability of creating increasingly sophisticated systems for weather simulations, better networking is essential for their improvement. Since networks and distributed applications develop quickly, research results can generally be expected to be picked up quickly and there are often new real-world aspects that need to be integrated. The timescales for this vary within the field, however. While mobile technology changes every six months and industry picks up and integrates new and even disruptive features quickly, the Internet's backbone is very stable and actual change needs a very strong research basis before its inertia can be overcome.

What Problems we Focus on, and How we Approach Them

The main characteristic of the ND department at the organisational level is its very tight relations with its sister research group at the University of Oslo. This is expressed in many ways: First, three senior professors at the university hold adjunct positions in Simula's ND department and three of the scientific staff of Simula hold adjunct positions at the university. But the collaboration runs deeper still. In Simula's first period (2001 through 2005) the distribution of senior personnel between Simula and the University of Oslo was different than it is today. At the beginning of the second period, there was a benign redistribution of researchers between the institutions. This redistribution led to the better utilisation of resources and a far stronger collaboration between the institutions than in the first period. The two groups share joint research projects as well as joint innovation initiatives and one of the Simula projects is currently administered by a professor from the university through his adjunct position at Simula. It is our firm belief that the long-term collegiality and openness between the two sister groups has been particularly important to the development of the ND department.

The aim of Simula Research Laboratory is to conduct long-term basic research with a clear view to application of the research results. This means that our projects cover the span from the pure basic research of fundamental properties of computer communication to close collaborations with industry, where the take-up of results is relatively quick. What all of these projects have in common, however, is that they started their life studying fundamental problems. One example is the project on interconnection networks (ICON). This activity started in the early 2000's and focussed on efficient and deadlock-free routing in arbitrary topologies. At the time, this was an unsolved problem that had no present-day application. As time proceeded, however, the internal network structure of supercomputers significantly increased in size, to the current situation where they contain thousands of switches. The mean time between failures is so small that routing algorithms that rely on the network being a regular structure are no longer feasible. Our research on interconnection networks therefore started as the study of esoteric network structures by means of simulation, but it has now developed into an activity that develops and tests routing algorithms on real hardware in collaboration with Sun Microsystems. More detail on this activity can be found in chapter 14.

At the other end of the spectrum is the Resilient Wireless Networks project that focuses on the fundamental properties of wireless transmission. Our research in this topic uses a combination of mathematical modelling, analysis, and simulations and the long-term goal is to contribute to resilient wireless communication services. Wireless resilience provisioning should take into account the unique characteristics and requirements in all protocols layers. In the physical layer, the main challenge lies in the unstable physical link; thus in this layer we emphasise channel modelling, signal processing, and cooperative communication. In the media access control (MAC) layer, resolving multiple access collisions is a major concern in achieving maximum throughput. In the routing layer, multipath routing with resilient coding schemes can be an option to delivery packets as much as possible with sufficient reliability and correctness. In the scheduling phase, transmitting packets in a fair and efficient manner is of the utmost importance. These are important issues that impact and shape the way future wireless devices will interoperate.

The RELAY project works on performance and resource utilization of time-dependent large-scale distributed systems. This project typically motivates its research from applications that stress resource usage linked to communication in one way or another. The team working on this project has had many years' experience conducting research in the area of video on demand, but at Simula they have extended their portfolio of applications to include highly interactive applications that are not necessarily bandwidth hungry. Examples of such applications are online games, audio conferencing, and financial systems and each of these imposes demanding restrictions on packet latency. The topics studied will typically have importance beyond their initial application area. It is in the nature of this research that it can, and maybe even should, be carried out in collaboration with partners who have direct interest in the application areas considered and the solutions to the problems attacked. A further description of the RELAY project can be found in chapter 17.

Finally there is the REsilient Protocols And Internet Routing (REPAIR) project, a research effort that concentrates on studying resilience in wired networks. The project is an answer to a request from the Norwegian Ministry of Transport and Communication for more research on sustained service on network infrastructures challenged by unexpected events. This project initially focussed on solving the problem of providing fast reactions to faults in IP networks and it started its studies using simulation models. Later a physical demonstration of the IP fast reroute solution was built and publicly shown at Middleware '07. To the best of our knowledge, this was the first public demonstration of IP fast reroute in the world. The company Resiliens was later formed in an attempt to commercialise the solution. More on the REPAIR project can be found in chapter 15.

The connection between all the projects in the ND group is that they all relate in some way to server-based distributed applications. This relation is, however, somewhat fictional, since tight relations between the projects were not among the guiding principles forming them. The projects were constructed from two criteria that we considered to be absolute:

- The potential for scientifically excellent results should be present. This means that to a large extent, topics were chosen so that we could build projects around our strongest researchers.
- The potential impact of the research should be high. This basically means two things: First, the projects should focus on problems whose solutions would have an impact outside academia. Second, each project should have a clear understanding of the processes that lead from a successful research result to application.

This second point does not mean that all projects should eventually end up in tight collaboration with industry. It does, however, mean that the research and researchers in the ND department should have metrics of success that span more than just the production of scientific papers.

Whats Next in Communication Research At Simula

The essential strategy of the ND department is to address fundamental problems of practical relevance and to publish research results at a high international level. The department strives for impact outside the research establishment. One way in which this was achieved was the establishment of joint long-term projects with industry in strategic collaborations, another the creation of spin-off companies based on research results. Yet another approach of high impact and visibility is contribution to important open-source projects: ND projects have contributed to OpenFabrics and Linux and aim at extending this approach. In addition, all of the ND department's projects are intensifying their use of experimental research. While this is highly important for the acceptance of publications by conferences at the highest level, it also provides a means of attracting industry interest through prototypes and demonstrators.

Thematically, the ND department continues to concentrate on the performance and reliability of networks and distributed systems. The scope will remain broad, ranging from networks on a chip, whose power consumption can be improved by new scheduling and routing decisions, to Internet backbone routers, where we extend our investigations into evolutionary approaches to increase the scalability of interdomain routing. Several projects intend to exploit cross-layer information to improve the efficiency of application-layer mechanisms over sublayers. Projects address the cross-layer theme in workload assignment to multicore chips, in routing over cognitive radio networks, and in coordinating overlay with network-layer routing.

Recent developments of research topics have led to a confluence of activities that is already showing in the cooperation of the ND department's projects. This natural, thematic overlap will increase cooperation and strengthen the department. The projects have different backgrounds but, through the department's strategy, will be able to cooperate naturally. For example, routing and job allocation research for networks on a chip in the ICON project meets workload partitioning on multicore chips in the RELAY project. Both projects are also investigating the data centre scale, where the research questions are similar to those on the small scale but with likely different answers. The external research project Verdione connects the SimTel, RELAY, and REPAIR projects in an effort to transfer time-critical high-bandwidth multimedia streams over long distances using resilient overlay networks. As a final example, the Resilient Wireless Networks project pursues a variety of local and cross-layer optimisation questions in cognitive radio networks and cognitive mesh networks in particular. In this research arena, they meet the SimTel project, which addresses the use of cross-layer information in cognitive radio networks in an experimental manner.

We expect that this cooperation will improve the homogeneity of the ND group and lead to better mutual understanding. They will deepen our understanding and allow a better concentration of our resources for greater impact in research and industry.

14

SCALABLE INTERCONNECTION NETWORKS

Olav Lysne, Tor Skeie, Sven-Arne Reinemo, Frank Olaf Sem-Jacobsen, and Nils Agne Nordbotten

Olav Lysne · Tor Skeie · Sven-Arne Reinemo · Frank Olaf Sem-Jacobsen
Simula Research Laboratory

Sven-Arne Reinemo
Sun Microsystems, Ltd.

Olav Lysne · Tor Skeie
Department of Informatics, University of Oslo, Norway

Nils Agne Nordbotten
Norwegian Defence Research Establishment

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_14,

PROJECT OVERVIEW

ICON – Scalable Interconnection Networks

A computer consists of several different entities, each of which performs highly specified tasks. Examples of such entities are units for storing information (e.g., disks and memory chips), units for computation (CPUs), and units that handle computation with the outside world. A computer is an orchestration of a set of such entities into a unified product. To support the collaboration between all of the entities that constitute a computer, there is an *interconnection network* that allows them to communicate. This interconnection network may be very small, like the one typically found inside a laptop computer, or it can be huge, like a supercomputer, which can contain thousands of switches and links. Thus, the interconnection network lies at the heart of the architecture of any computer, and the performance and reliability of the computer depends heavily on the performance and reliability of the interconnection network.

Industrial activity in interconnection networks has been occurring in Oslo since the early 1990s. It began with a commercial company called Dolphin Interconnect Solutions, which developed and marketed a technology called SCI, it has been continuous ever since. An illustration of Oslo's standing in the field is that Sun Microsystems chose it as the home for its new facility to develop components for the interconnection technology called Infiniband. About 50 per cent of the funding for interconnection research at Simula has come from the Simula base funding; the rest has come from the European Framework Programmes and from the Research Council of Norway. Recently a contract was signed between Sun Microsystems and Simula for a three-year jointly funded research project.

Scientific Challenges

The main scientific challenge addressed to date is how to deal with interconnection networks that are so large and contain so many components that failing hardware must be considered a part of normal operation. Because interconnection networks of these magnitudes reside in computers that cost millions of dollars, the main research approach has been to simulate them with our own custom built simulators.

Obtained and Expected Results

The main scientific results obtained to date are: efficient routing methods for interconnection networks that contain faults and effective methods for reaction at the moment when the fault occurs. We have published these results in leading journals. Moreover, some of our methods have been adopted by Dolphin, Silicon Graphics, and Sun Microsystems, and they now run on computers that are in daily production. Our implementation of some of our results in Open-source also has been a very efficient way to promote our results.

SCALABLE INTERCONNECTION NETWORKS

14.1 Introduction

A modern supercomputer or large-scale server consists of a huge set of components that perform processing functions and various forms of input/output and memory functions. All of the components unite in a complex collaboration to perform the tasks of the entire system. The communication between these components that allows this collaboration to take place is supported by an infrastructure called the *interconnection network*.

Interconnection networks typically fall into one of two categories. The first, which is commonplace in small systems, is shared media buses. Shared media buses, however, are severely limited in scalability, as only one message can be sent across the bus at a time. For larger systems, the interconnection network is constructed from links coupled together with switches. Current technologies for such networks include Gigabit Ethernet [4], InfiniBand [63], Myrinet [8], and Quadrics [7].

Although there are many conceptual similarities between interconnection networks and the wired infrastructure of the Internet, research in these two domains has been largely separate. Different groups of researchers work in the two areas and results are published in different journals and presented at different conferences. While this can be partially explained by how the fields emerged, the main reasons are related to profound technical differences that limit the extent to which a result from one field is applicable to the other.

Interconnection networks typically exercise link level flow control, meaning that no switch is allowed to drop data packets. This feature implies that a packet will be prevented from proceeding when no buffer space is available at its next intermediate hop. A situation where no packets can proceed is defined as *network deadlock* and routing strategies for interconnection networks must be carefully designed so that this situation does not occur. An Internet routing strategy will most probably result in a deadlock when utilised in an interconnection network. Furthermore, an efficient routing strategy for interconnection networks will most likely not compete with strategies that do not need to worry about deadlocks and for these reasons research results do not easily cross the border between these domains.

Another profound difference lies in their network topologies. In interconnection networks, research on topologies consists of the development and study of regular structures that are to be built into a single machine. Research on Internet topologies must, on the other hand, consider topologies that grow organically as society evolves. Regarding traffic patterns, self-similarity (the traffic patterns display similar characteristics on both short and long time scales) for the Internet differs from the patterns generated by single applications in an interconnection network. Finally there is the effect of congestion: Whereas a congested link in the internet leads to packet drops, packets are not dropped in interconnection networks; therefore congestion spreads in a tree-like fashion from the congestion point.

Research in interconnection networks has gone through a series of phases. It was initiated by the seminal work by Dally on the torus routing chip [59]. The novel aspect at that time was that technology had advanced sufficiently for it to be possible to integrate a switch onto a single silicon chip[1]. This work started a period during which progress was made on routing strategies, deadlock control, topologies, and fault tolerance. In the late 1990s, however, many researchers abandoned the field. There were mainly two reasons for this: First, the number of machines that were built needing really large interconnection networks was limited to a very small niche in high performance computing (HPC). Second, even in these systems the interconnection network was not seen as the bottleneck.

In the early 2000s, Simula Research Laboratory started its focussed research on scalable interconnection networks by initiating a project named *ICON*. Two important observations formed the basis for this decision. First, an extrapolation of the technical development at the time implied that the process of increasing computer speeds by increasing CPU clock frequency that had served the industry well over decades was coming to an end. The obvious way to further increase computing power was to allow sets of processor cores collaborate in solving problems in parallel. These cores would need an interconnection network to collaborate and therefore these networks found their way into mainstream computing. The second observation was that the state of the art at the time was to view the interconnection network as a fixed regular structure. Since the quest for greater computing power in the supercomputing segment had to be met by an increase of several magnitudes in the number of cores, this would require interconnection networks of sizes that were unsustainable in the rigid view of the network as the expected mean time between component failures got unacceptably low. This meant that research on the flexibility of such networks was necessary.

In this context ICON focuses on the science and technology of how to connect links and switches into scalable network topologies and how to route packets effectively in these networks so that they yield the highest possible performance. The development of stable, efficient interconnection networks is an important element in order to cope with the increased burden imposed on the interconnection network by modern data centres and in HPC clusters. Some of the results of this project have had a major impact on the routing architecture of modern supercomputers (see section 14.5).

In particular, ICON focuses on topics related to the following:

- Topology-agnostic routing: new deadlock-free strategies for routing non-regular interconnection networks.

[1] This also required the invention of wormhole routing, where no switch needed to have buffer space for an entire packet.

- Fault tolerance: the ability of the interconnection network to be resilient to components failing without warming.
- Dynamic reconfiguration: the ability of the interconnection network to change routing functions without entering into deadlocked states in the transition phase.

Sections 14.2 through 14.4 elaborate on these three topics in order to better present their importance and relevance. Section 14.5 covers the industrial adoption of the results. Section 14.6 explores the future outlook of the field before the conclusions in section 14.7.

Background

In the last decade the size of cluster-based computers has increased dramatically and has surpassed proprietary supercomputers in popularity. The November 2008 list of the top 500 supercomputer sites [9] includes several clusters with more than 10,000 processors, indicating that a paradigm shift towards clusters has taken place in the HPC landscape. Currently, there are several high-performance commercial off-the-shelf (COTS) interconnection network technologies available that are used to build HPC clusters, such as Gigabit Ethernet [4], InfiniBand [63], Myrinet [8], and Quadrics [7]. The adoption of cluster-based supercomputers has also led to the increased use of such systems outside of traditional scientific communities.

The interconnection networks used by HPC clusters are switch-based networks whose topology is defined by the customer. The topology can be either regular or irregular. Regular topologies such as direct topologies and multistage networks are often used when performance is the primary concern. For instance, the first machine in the top 500 supercomputers list, the IBM Roadrunner at the Los Alamos National Laboratory, has 130,464 processor cores interconnected by a COTS InfiniBand network with a fat tree topology. The second system on the list, the Cray XT4 Jaguar system at Oak Ridge National Laboratory, has 150,152 cores interconnected by a Cray SeaStar proprietary network with a three-dimensional torus topology. Almost all machines on the top 500 list use some form of regular topology, but irregular topologies are indirectly present because regular topologies become irregular when components fail. Therefore, topology-agnostic routing algorithms are crucial for the utilisation and performance of such systems.

Considering reliability from an HPC cluster and communication perspective, the ability of the network to provide sustained service under the failure of single components is of vital importance. This is driven by the fact these installations are expensive to build and operate and they have a limited lifetime due to Moore's law. Therefore, utilisation should be maximised by having the system running 24 hours a day, seven days a week, even in the case of faults. As systems become increasingly larger, the number of interconnected components grows and the probability for faults at any given time increases. Faults in the interconnection network may potentially leave the remainder of the system disconnected; thus, providing fault tolerance in interconnection networks is essential to the system's overall reliability and utilisation.

As stated previously, agnostic methods have the ability to always provide a connected and deadlock-free routing function as long as the network is physically connected, but it is only one component in the step towards a fault-tolerant interconnection network. How to handle the fault when it happens requires a more complex fault tolerance mechanism and many such mechanisms have been proposed. Some solutions rely on specific logic (i.e., adaptive routing) not present in current COTS technologies or on disabling some regions of the network, including healthy nodes [56], thereby reducing the utilisation of the network. Other solutions rely on duplication of hardware or are based on static or dynamic reconfigurations of routing tables. With dynamic reconfiguration it is possible to reconfigure the network without interrupting application traffic, as opposed to static reconfiguration, where application traffic is stopped during reconfiguration. Therefore, dynamic reconfiguration is important as a fault tolerance mechanism because it minimises the impact of faults on active applications. Furthermore, dynamic reconfiguration is also useful for the virtualisation of large compute clusters that are commonly divided into multiple virtual servers. In order to control the sharing of interconnection resources between the virtual servers, it is beneficial to be able to change the routing function so that each server becomes routing contained, that is, they do not share physical resources in the network.

Basic Concepts

Three issues dominate the design of an interconnection network: topology, switching technique, and routing algorithm [2]. The switching technique specifies how the network resources are allocated to packets and what happens to those packets that are waiting on a busy resource. In lossless networks, packets waiting to acquire a network resource are buffered. This buffering can be carried out either in units of packets, as in store-and-forward and virtual cut-through routing, or in smaller units of data commonly referred to as flits, as in wormhole routing [67]. Unlike store and forward, virtual cut-through and wormhole routing allow transmission over each hop to be started as soon as the required resources are allocated, without waiting for the entire packet to be received. Thus, in the absence of blocking, packets are effectively pipelined through the network. But when the buffer associated with a certain physical channel is full, no other packet can access the physical channel, even if the packet is not blocked. This can be avoided by having several *virtual channels* (VCs) multiplexed across the physical channel and the buffer allocation for each VC decoupled from the physical channel. Each VC is implemented with an independently managed Buffer, which prevents a packet that is buffered from blocking the physical channel [60].

An issue closely associated to the switching technique is flow control. An inherent property of lossless networks is that they shall not lose packets in transit when buffers become full. In general, flow control is a mechanism for reporting the current availability of buffer space at the downstream end of the channel and whether to allow the next packet to be transmitted. When buffers become full; no more credits are available, the flow control acts as a back-pressure mechanism that possibly restrains packets throughout the network in order to avoid packet loss. When VCs are used, flow control is applied on a per-VC basis. The flow control mechanisms commonly used by current network technologies are *credit-based flow control* for InfiniBand and *stop and go* for Myrinet and Gigabit Ethernet.

In order to efficiently forward packets through a network, a routing algorithm must be implemented. Routing methods can be either deterministic or adaptive. In deterministic routing the path taken between a given source-destination pair will always be the same. This is achieved by the switches providing only one routing option (output port) for a packet. Adaptive routing may offer several routing options (paths). The selection of the routing option is usually based on the current occupancy status of the links. In that respect, adaptive routing algorithms may avoid congested areas, thereby boosting performance. Although adaptive routing has demonstrated its superior performance compared to deterministic routing, of the existing COTS technologies, only Quadrics supports adaptive routing. InfiniBand, Myrinet, and Ethernet rely only on deterministic routing. This is mainly because adaptive routing does not guarantee in-order packet delivery, which is a prerequisite for many applications.

Another key issue in the design of routing algorithms is how to prevent deadlock and livelock. *Deadlock* occurs when no packet can advance towards its destination because the requested network resources (buffers/channels) are not released by the packets occupying them. This occurs when waiting packets form cyclic resource dependencies. Deadlock freedom in the network is guaranteed if there are no cycles in the resource channel dependency graph. Conventional graph theory can be used to prove this. Avoiding cycles can be achieved by imposing some restrictions on the routing, e.g., enforcing an ordering in the allocation of the resources as in dimension-ordered routing [60]. Also, VCs can be used as a technique to avoid cycles in the dependency graph [60]. In this context, some adaptive routing schemes allow for cycles in their resource graphs while still remaining deadlock free. This requires that they have a routing subfunction (an escape path) that is free from cycles [3]. *Livelock* occurs when packets are allowed to advance but never reach their destination. Livelock typically happens when non-minimal routing is used. It can be avoided by bounding the number of misroutings that a packet may take, thus ensuring packet progression. Alternatively, livelock can be handled in a probabilistic manner by assuming that the probability for a packet remains in the network for a time t that approaches zero as t tends to infinity.

The irregularity of a topology induced by one or more faults or by the design of the topology makes routing more complicated. For instance, dimension-ordered routing (DOR) is a dedicated/specialised routing algorithm proposed for meshes and tori networks. This routing algorithm forwards every packet through one dimension at a time, following an established order of dimensions. Therefore, DOR is not able to route packets even in the presence of a single fault.

Run-time faults in interconnection networks can be dealt with in several ways, as previously mentioned, such as deploying redundant components, globally or locally redundant paths, and through the global or local reconfiguration of the network. Within all these solution domains ICON has made contributions. We will discuss the different strategies in more detail below, but let us first introduce the two main fault models used in this community. Faults can be handled either statically or dynamically. The fault model used may have significant implications on the system-level design, with respect to how large a part of the system must be shut down in the presence of faults, the complexity of the repairing system, and so on.

In a static fault model the network is shut down when failures occur in order to reestablish routing connectedness followed by a system restart for continued operation. Basically, when connectivity must be reestablished, the redundant paths in the network are used to route around the faulty areas. Considering that the mean time between failures of a system may be lower than the execution time of some of its hosted applications, a static fault model might need to be combined with a higher-level checkpoint protocol, so that the application can be rolled back to the last checkpoint before the fault occurred rather than having to restart after the reconfiguration. During this repairing process the network will be emptied. Depending on the frequency of the checkpointing, the application will lose some of the accomplished work in progress. Such a static fault model repair process may be too costly from an application downtime perspective and therefore not acceptable.

More recently we have seen a shift towards a dynamic fault model. When such a model is used, the system remains operational in the presence of faulty components. This can be achieved by letting the switches have built-in intelligence to circumvent the failures through the rerouting of packets or by letting the network be dynamically reconfigured by a management agent. In a dynamic fault-model-based system the repairing process is quite local. The applications are to a lesser degree affected by faults, though a small drop in performance may be observed due to the retransmission of lost packets and reduced network bandwidth. Compared to a static model, however, more sophisticated network components that implement various, possibly complex, fault tolerance concepts will typically be needed.

14.2 Topology-Agnostic Routing

The first routing algorithms for interconnection networks were developed for specific fixed topologies of regular shape. These included e-cube routing for hypercubes [53], xy routing for meshes, and routing based on the ordering of VCs for tori [60]. Two developments did, however, create a need for routing strategies for irregular topologies. The first was the introduction of off-the-shelf switching components for clusters of workstations. These components needed to handle any pattern of connections that the user might create. The second important development was a growth in system size that spawned a need to route topologies with one or more faulty components.

A plethora of routing strategies have been developed for irregular networks. One of the first to be used in cluster computing was the up*/down* (UD) algorithm [70], which quickly gained popularity for its simplicity. The algorithm performs a breadth-first search from a root node, denoting one direction of each link as *up* (towards the root) and the other direction as *down*. The up direction of links form a directed tree towards the root and the down direction form a directed tree away from the root. The only allowed packet routes first traverse a sequence of links upward, followed by a sequence of links downward. Cyclic dependencies between links are therefore avoided by disallowing any message to traverse a down link followed by an up link.

Further refinements were proposed, such as the use of a depth-first spanning tree [52] and the flexible routing scheme [69], which improved traffic balance by breaking the cycles in each direction at different positions. Another algorithm based on a UD spanning tree, the left-up first turn routing algorithm [51], uses a left-to-right directed graph, distributing the traffic around the root node of the spanning tree and achieving better network balance.

The main shortcoming of the UD-based routing algorithms is their scalability. There is a severe tendency for the network to become congested around the root. Other approaches have been proposed to alleviate this problem. The segment-based routing algorithm [50] uses a divide-and-conquer approach, partitioning a topology into subnets, and the subnets into segments, and placing bidirectional turn restrictions locally within each segment. Placing turn restrictions results in a greater degree of freedom when compared to the previous routing strategies that rely on heuristic rules. Adaptive-trail [68], and smart routing [57] achieve significant performance improvements. Their applicability is, however, limited, because the first requires extra functionality not usually present in the switches and the latter has a very high computational cost. In 2001 we proposed an improvement of UD routing by using VCs [66] to divide the physical network into a set of virtual networks (layers). The method is further based on having one UD tree in each layer, each placing its root in a different place in the topology (MROOTS). This eliminates much of the traffic congestion around a single root and results in a significant performance improvement compared to conventional UD routing.

Neither MROOTS nor the other routing strategies discussed previously guarantee that all packets will be routed through minimal paths. This can lead to increased packet latency, especially for short packets. Use of non-minimal paths may also increase overall link utilisation, resulting in poor performance and scalability. In 2002 we presented a new topology-agnostic routing method that could guarantee shortest-path routing [71]. In the same way as MROOTS, it assumes VCs (layers). It works by assigning subsets of the minimal paths between the source-destination pairs to different virtual layers so that each virtual layer is deadlock free and therefore deadlock freedom of the resulting routing function is guaranteed. The method is called LASH, for LAyered SHortest path, and it outperforms UD and MROOTS with a cost of only a modest number of virtual layers.

The following section describes LASH in some detail. Industrial adoption of this routing strategy is discussed in section 14.5.

LASH

The objective of this research was to devise a topology-agnostic routing method that could guarantee shortest-path routing. In general, the deadlock problem will often disallow shortest-path routing in an irregular topology. As a result, most existing methods for shortest-path routing in irregular networks provide shortest paths only relative to some constraint. An example of this is UD routing, as defined previously, in that it supports shortest paths only relative to the constraint that no up channel can be used after a down channel.

LASH solves this problem by using layered routing to allow for true shortest-path routing in irregular topologies. This is done by first finding a shortest physical path between every source and destination. Thereafter, we assign the <source, destination> pairs (the minimal paths) to different layers in such a way that that all <source, destination> pairs are assigned to exactly one virtual layer. In addition, we make sure that each virtual layer is deadlock free by ensuring that the channel dependencies stemming from the <source, destination> pairs of each layer do not generate cycles. The routing function R is from that point of view defined by two subfunctions, R_{phys} and R_{virt}, respectively. The former defines one minimal physical path for each <source, destination> pair. The latter determines on which virtual layer packets from each <source, destination> pair should be forwarded. Below we give an algorithm that assigns <source, destination> pairs to virtual layers.

Step 1: Obtain R_{phys} by finding the shortest path between all <source, destination> pairs within the network.

Step 2: Take one <source, destination> pair *sd* that has not yet been assigned to a virtual layer. Find an existing virtual layer vl_i such that *sd* can be added to R_{virt_i} without closing a cycle of dependencies in virtual layer vl_i. Add *sd* to R_{virt_i} (basically, this step verifies that virtual layer vl_i remains free from deadlocks).

Step 3: If step 2 is unsuccessful, create a new virtual layer vl_j and let R_{virt_j} contain only *sd*.

Step 4: If there are more <source, destination> pairs that have not yet been assigned to a virtual layer, go to step 2.

Step 5 (optional balancing step): Find a <source, destination> pair that has been assigned to vl_{max} (the virtual layer hosting the most <source, destination> pairs) and move it to vl_{min} if it does not close a cycle within vl_{min}. Repeat this step until the number of <source, destination> pairs is equally distributed between all the virtual layers or balanced as equally as possible.

Step 6: Obtain R_{virt} from the sets R_{virt_i} for all vl_i by letting all packets from each <source, destination> pair in R_{virt_i} be routed on virtual layer vl_i.

An important issue in the evaluation of layered routing is the number of layers that are needed to grant shortest-path routing to every <source, destination> pair. The required number of layers depends on both network size and connectivity.

Some answers are easily derived. If we have minimal connectivity so that the network has the shape of a tree, one virtual layer suffices, because no cycle of dependencies can be closed as long as all routing follows a shortest path. Furthermore, networks with maximal connectivity will also need only one layer, because no packet will traverse more than one link and so no channel dependencies will exist. Most of the network topologies that are used in practice, however, will fall somewhere in between these two extremes and in order to evaluate the relation between network connectivity and the need for virtual layers, we conducted a series of experiments. We considered two network sizes of 16 and 128 nodes, respectively. For each of these sizes, we considered a range of connectivities from minimal connectivity and upwards towards maximal connectivity, adding one or two links at a time. For each network size and connectivity, we generated 100 random topologies and subjected them to a modified version of the algorithm above, which is always able to provide another virtual layer on an as-needed basis. Figure 14.1 shows the average, maximum, and minimum number of virtual lanes needed in each case.

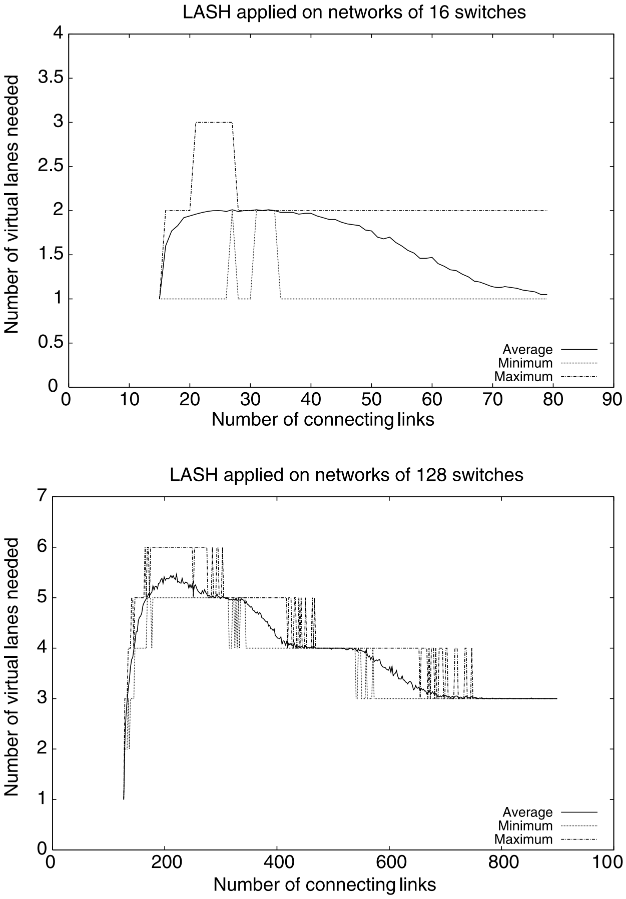
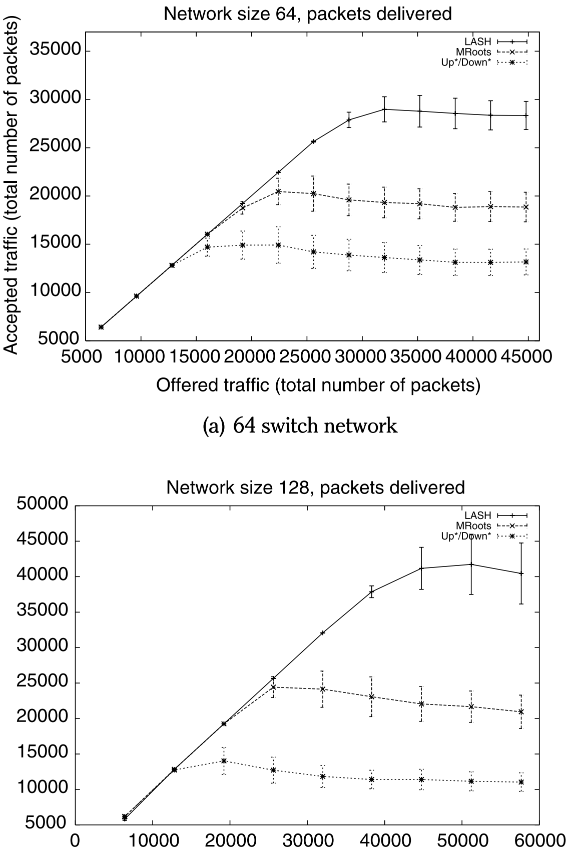


Figure 14.1 Number of layers needed to allow shortest paths between all <source, destination> pairs in LASH on networks with a varying number of switches.

When it comes to regular networks, the need for virtual layers is independent of the size but, on the other hand, may depend on the network's number of dimensions. Furthermore, it will depend heavily on the choice of physical paths. For example, for a fault-free mesh and a hypercube, LASH requirements can get as low as a single virtual layer, even for higher-dimension networks. Two-dimensional tori networks need three layers, while three-dimensional tori require five. The reason for this is that tori networks contain a ring in each dimension that is more prone to deadlocks.

Being able to route minimally is highly beneficial with respect to network performance. Figure 14.2 shows the throughput of LASH compared to MROOTS and UD for two irregular networks consisting of 64 and 128 switches, respectively, where the evaluation has been conducted by simulations under equal conditions (e.g., using the same number of virtual layers, including deterministic routing). What we see is that MROOTS performs significantly better than conventional UD and that LASH again outperforms MROOTS, with the gap between the three methods increasing with network size, since neither UD nor MROOTS can guarantee shortest-path routing. For further details about the performance figures of MROOTS and LASH we refer to [49, 48].



(a) 64 switch network

(b) 128 switch network

Figure 14.2 Throughput of UD, MROOTS, and LASH under uniform traffic for random generated irregular networks.The error bars show the 95% confidence intervals.

Adaptive LASH

Originally LASH was designed as a deterministic routing method. This was motivated partly by the fact that the major cluster technologies, such as InfiniBand and Gigabit Ethernet, only support deterministic routing. It is also partly related to the HPC community's preference for simple switching logic and in-order delivery of packets[2]. Recently, however, there have been a few initiatives indicating that the HPC community is changing its view on this.

The LASH method can easily be enhanced to take advantage of adaptive routing features. As already discussed in [49], two adaptive versions of LASH are possible. One is where we allow for adaptivity in the source nodes, which we call multipath LASH (MP-LASH). In most networks, there will be more than one shortest path between any source and destination. In fact, the InfiniBand standard allows for multiple paths between any source-destination pair, by offering multiple virtual addressees for each physical destination, where each of them identifies a different path through the network. The source node can then choose which path to use, depending on the conditions in the node. The added freedom can be exploited within the LASH framework basically by changing step 1 in the algorithm to consider all the shortest paths between any pair of nodes. This will, of course, have an effect on the need for virtual layers compared to pure deterministic routing. Our experiments reported in [49] show that this effect is very limited.

The other approach to utilising multiple paths between each source-destination pair is to introduce adaptivity in the switches. This allows each switch to choose between different output ports on a per-packet basis. The problem herein is that we do not know in advance which path a packet will follow. Therefore, the algorithm for generating the routing function must cater to a non-deterministic choice of paths. This implies that the algorithm assigning paths to layers cannot treat more than one path at a time. It must consider all the paths between a pair of source and destination and assign all of these to the same layer. As a consequence, the need for virtual layers increases more than for MP-LASH. A full discussion of this is given in [49].

LASH in OpenFabrics and on InfiniBand

The main limiting factor on the impact of a theoretical study on computer architecture is that the hardware needed to take advantage of the development may never be built. Routing studies in interconnection networks are no exception, in that the proposed solutions will need some features in the host adaptors and/or the switches. In the case of LASH, however, there is a very good match between the properties of the proposed method and the properties of an established industry standard called InfiniBand [63]. Over the recent years InfiniBand has increased its market share in HPC. InfiniBand currently has 25 per cent of the market and this trend appears to be continuing.

In InfiniBand switches make the physical routing decision only based on the current switch and the destination (known as $N \times N \rightarrow C$ routing). The virtual

[2] Adaptive routing may result in out-of-order delivery, which requires costly housekeeping that may harm performance.

layer[3] (*VL*) is determined from a service level (*SL*) tag included in the packet header. This is called the $SL \rightarrow VL$ mapping. Given these routing properties, implementing LASH is straightforward: R_{phys} maps directly to the $N \times N \rightarrow C$ routing attribute of InfiniBand, indentifying a single outlink per destination on each switch. Also R_{virt} can be implemented by the $SL \rightarrow VL$ mechanism as a one-to-one mapping; recall that the packets never switch between the virtual layers. Thus, for implementing LASH in InfiniBand the need for SLs follows the required number of virtual lanes.

The increasing popularity of InfiniBand with multiple vendors has made the OpenFabrics Alliance see the light of day [47]. This is an open-source initiative that aims to develop and promote InfiniBand protocol stack software and management functionality. Most of the key HPC suppliers stand behind the initiative, having made it to a successful HPC open-source community. In 2006 we contacted OpenFabrics to check their interest in LASH. As a result, a dialogue was established with the community and the following year we made LASH available in the OpenFabrics software framework. It was later used by Silicon Graphics and is to be used by Sun Microsystems/Sandia National Laboratory, as is discussed in section 14.5.

14.3 Fault Tolerance

Topology-specific routing algorithms such as DOR, used in mesh and tori networks, do not offer fault tolerance properties in their original form, since they are not able to utilise the redundancy provided by the network topology. Topology-agnostic methods, on the other hand, have the capability of providing connectivity as long as the network remains physically connected. But the agnostic algorithms do not offer fault tolerance per se; they either have to be combined with a reconfiguration of the network (i.e., calculate a new routing function/paths and download new routing tables accordingly) or redundant paths must be computed (embedded in the original routing table as alternative paths) relative to the constraints given by the agnostic methods. In light of this, much work has been done to create specific fault-tolerant routing algorithms that can take better advantage of the redundancy offered by a certain topology.

One approach has been to develop adaptive routing algorithms, where the adaptivity can be used to circumvent faulty components and in this way provide dynamic fault tolerance. Notice that adaptive routing algorithms are not necessarily fault tolerant. A strictly minimal adaptive routing algorithm is not able to handle a single fault, since the source-destination pairs connected by a single minimal path are disconnected by any fault within this path. Linder and Harden [64] proposed a method providing sufficient adaptivity to tolerate at least one fault in mesh and tori networks. The number of VCs, however, required by their method increases exponentially with the number of dimensions. There are also improvements of Linder and Harden's concept based on the observation that the high number of VCs required is due to the freedom to traverse dimensions in an arbitrary order. For example, Chien and Kim proposed planar adaptive routing [58], where adaptivity is limited to adaptive

[3] Virtual layer is the InfiniBand term for virtual channel.

routing in two dimensions at a time. This method requires at most three VCs for meshes of any dimension but does not properly handle faults on the edges of the network.

Glass and Ni [62] used the partial adaptivity provided by the turn model [61] to develop a fault-tolerant routing algorithm for meshes. Their method does not require any VCs, but only tolerates $n-1$ faults in an n-dimensional mesh and uses non-minimal paths in the fault-free case. The turn model is also utilised by Cunningham and Avresky [46], who provide fault-tolerant routing in two-dimensional meshes using two VCs. Their method incurs a significant performance loss by a single fault, however, as adaptive routing must be disabled. It also requires healthy nodes to be disabled.

Boppana and Chalasani [54] use local information to create rectangular fault regions in two-dimensional meshes with DOR. The nodes and links on the border of a fault region are non-faulty and create an *f-ring* or *f-chain* used for rerouting packets around the fault(s). By combining this method with planar adaptive routing, it can also be applied to higher-dimensional meshes. An improved version by Sui and Wang [45] is able to tolerate overlapping fault regions in meshes using three VCs. Using rectangular fault regions has the disadvantage of disabling an unnecessarily high number of healthy nodes. Kim and Han [44] partly address this issue by extending the method to support overlapped non-convex fault regions in meshes, using four VCs. Recently, Gu et al. [43] proposed extensions to also support concave fault regions. This latter method can be applied in combination with previous proposals for handling non-convex faults in meshes and tori. It requires ejecting and reinserting packets when entering/leaving a concave section, however, thereby increasing latency and occupying memory at the nodes. Park et al. [42] handle simple concave, non-overlapping fault regions in meshes without ejecting/reinserting packets, requiring three or four VCs, depending on the provided fault tolerance. This method does not, however, handle faults on the edges of the mesh.

Chalasani and Boppana [41] also proposed a variation of their method for the torus requiring a total of six VCs. Shih later improved on this by proposing a method tolerating block faults in tori using three VCs [39] and another proposal tolerating non-convex fault regions when using four VCs [40]. Unless, however, combined with the method of Gu et al., where packets are absorbed and reinjected when entering a concave region, these methods may require healthy nodes to be disabled. On the positive side, the topology-specific fault-tolerant methods discussed above all support a dynamic fault model, but on the negative side they have limited fault tolerance capabilities (handling only a few faults and possibly also disabling healthy nodes).

Simula's contributions in this area consist of several elements. Together with a group from the Technical University of Valencia, we proposed a fault-tolerant routing methodology based on routing packets via intermediate nodes [38]. This methodology supports fully adaptive routing and is able to tolerate any combination of five faults in three-dimensional tori without disabling healthy nodes. It is well known that two VCs are required in tori when used in combination with bubble flow control [37]. Recently, we proposed a fault-tolerant methodology for mesh and tori networks that is able to combine support for a dynamic fault model, tolerates multiple faults, and offers fully adaptive routing, while at the same time not requiring global reconfiguration or stopping packet injection at any time, in a fully distributed manner using a limited number of VCs [36]. The methodology requires no VCs in meshes, three VCs in two-dimensional tori, and four VCs in three-dimensional tori. For all the topologies, fully adaptive routing can be supported by adding at least one additional VC. Further, the method tolerates faults on the edges of the network and is able to handle overlapping concave fault regions without absorbing and reinjecting packets.

Below we will describe some of ICON's fault-tolerant contributions in more detail.

FRoots: Fault-Tolerant and Topology-Flexible Routing

The objective of this work was to devise a method providing dynamic fault tolerance for arbitrary networks without requiring complex functionality in either the switches or the end nodes. The method is novel in that it uses redundant paths; that is, if the network remains physically connected, there is a legal path for every source-destination pair in the presence of a single fault. FRoots can therefore allow continuous network access and, furthermore, can be designed to deliver all uncorrupted packets. If the network happens to not be connected after a fault, it can be used on each subnetwork.

The main idea of FRoots is related to the following observation: Assume a network with UD routing and consider a node whose connected links all have their up direction going away from the node. Let us call this node a *leaf*. The leaf will have no bypass traffic, because any traffic passing through it would have had to have made an illegal transition from a down channel to an up channel at the leaf. Therefore, if a leaf dies, all other nodes (switches) will still be able to communicate. Only traffic destined for or originating at the leaf is affected; the former cannot be delivered anyway and the latter will no longer be generated, since the leaf is dead.

In FRoots the idea is to use VCs to partition the network into a number of layers. Furthermore, each layer is assigned an individual, deadlock-free UD graph in such a way that all nodes are leaves in at least one layer. This allows FRoots to guarantee redundancy for single faults. FRoots uses the algorithm in Figure 14.3 to iterate over the network, attempting to make all nodes a leaf in at least one layer.

Guaranteeing redundant legal paths between any source-destination pair does not, on its own, suffice to cope with multiple faults nor does it support intended network changes. Multiple faults (faults or insertions) can, however, be handled by dynamic reconfiguration of the network, assuming they occur as a series of single faults or single changes separated by at least the time it takes to reconfigure the network.

Irregular	16-2	16-3	16-4	32-2	32-3	32-4	256-4	1024-4	16K-4
Maximum	6	7	8	6	7	8	8	8	8
Average	4.2	5.1	6.2	4.3	5.2	6.1	6.6	7.0	7.2

Table 14.1 Layers needed for various network sizes (N-L stands for nodes-links/nodes).

```
· Create a copy $\mathbb{K}$ of the network topology.
· Create a set notleaf initially consisting of all nodes in $\mathbb{K}$.
· While notleaf is non-empty:
    · Find a previously unused layer $\mathcal{L}$ in the network
    · Choose a node $\gamma$ in notleaf as $\mathcal{L}$'s guaranteed leaf.
    · Remove $\gamma$ from $\mathbb{K}$.
    · Create an empty set leafcandidates.
    · Move $\gamma$ from notleaf to leafcandidates.
    · Find the articulation points of $\mathbb{K}$.
    · For each node $\delta$ in notleaf where $\delta$ is not an immediate neighbour of any
    node in leafcandidates and $\delta$ is not an articulation point of $\mathbb{K}$:
        · Remove $\delta$ from $\mathbb{K}$.
        · Find the new articulation points of $\mathbb{K}$.
    · Create a UD graph of $\mathbb{K}$ as normally, except ignoring dangling links (links
    · previously connected to the nodes of leafcandidates)
    · The dangling links' directions are set to have the up direction from the
    nodes of leafcandidates to the nodes in $\mathbb{K}$.
    · Add all nodes of leafcandidates back to $\mathbb{K}$.
    · Copy the link directions of $\mathbb{K}$ into layer $\mathcal{L}$.
```

Figure 14.3 Algorithm to calculate FRoot graphs.

An important metric of FRoots is its need for VLs in order for all nodes to be a leaf in at least one layer. Table 14.1 shows the number of layers needed to guarantee one-fault tolerance in irregular networks of different sizes. Since the selection of the guaranteed leaf is random for each layer, our algorithm does not necessarily find the minimum number of layers needed. Still, table 14.1 shows that the number of layers needed is modest and scales very well; in fact, the maximum of eight does not change at all from 16 nodes/64 links to 16K nodes/64K links. Eight layers are possible to implement in hardware and in InfiniBand[63] 15 VCs are available for data (although these are intended for quality of service purposes, not for fault tolerance). An interesting observation is that the number of average layers needed increases more when the links-to-nodes ratio increases than when the number of nodes increases. Note that after one fault has happened, there is no guarantee that every node still has a "safe" layer to ensure two-fault tolerance, although most nodes will have one. Increasing the number of layers beyond these maximum figures will alleviate this problem. Another solution is to reconfigure the network after the first fault so that the network is able to handle the second fault. For a more detailed discussion of this we refer to [33].

Interconnection networks and Internet networking have profound technical differences, as discussed in the Introduction, but the problems addressed by the two communities sometimes have similarities. At Simula we developed concepts from FRoots into a set of mechanisms for fast rerouting in IP networks [32]. Fast rerouting around faults in intradomain IP routing has been a researched topic in this community for years and the wide applicability of the concepts from FRoots in both communities is a relatively rare exception. For further detail we refer to chapter 15.

Dynamic Fault Tolerance in Fat Trees

Fat trees have become more frequently used in supercomputers recently because of their good performance characteristics and the increase in switch radix in modern interconnection network technologies. A fat tree has multiple roots and belongs to the multistage interconnection networks (MIN) class of topologies. The ICON project has contributed dynamic fault-tolerant solutions for fat trees that are based on dynamic local rerouting around the faults and that work for both deterministic and adaptive routing [35]. Previous approaches to dynamic fault tolerance in fat trees (MINs) rely on strict routing rules, need extra mechanisms in the switches and additional network resources, and may have to forward packets through several routing passes via intermediate nodes [34].

The objective of this research was to devise a fault-tolerant methodology for fat trees that relies on local (dynamic) rerouting. A fat tree is seen in figure 14.4. The method requires only simple misrouting functionality in the downward routing phase (i.e., from a common ancestor node of the source and destination towards the destination) and a small modification of the port-selecting function in the upward routing phase. It tolerates $(radix/2) - 1$ arbitrary simultaneous link faults, where the radix is the size of the switch (number of ports). To the best of our knowledge, this was the first reported method for fat trees to be based on such principles.

The fault-tolerant routing algorithm relies on the numerous paths between two neighbouring switches. When a link between two switches fails, the switch at the top of the failed link will no longer have a path to the destinations it normally used to reach through the failing link. A nice property of the common implementation of fat trees (k-ary n-trees [10] and m-port n-trees [11]) is that there are multiple paths between two switches at neighbouring switching tiers in addition to the shortest-path link. These paths are two hops longer than the shortest path and consist of rerouting the packet down through another non-faulty link and then back up one tier to a different switch. The path followed by a packet that encounters a link fault in both the upward and downward phases is displayed in figure 14.4. The switch down which the packet is rerouted becomes a U-turn switch, a switch where a downward to upward transition takes place. These transitions are not allowed in regular fat tree routing, so we must impose some rules to avoid deadlock. These will be explained after we present the general routing algorithm to tolerate $(radix/2) - 1$ arbitrary link faults.

1. In the upward phase towards the root, a link is provided by the forwarding table as the packet's outgoing link. If the link is faulty, it is disregarded and one of the other upward links is chosen by the rerouting mechanism, rerouting the packet.
2. In the downward phase (towards tier $n-1$), only a single link is provided by the forwarding table, namely, the link on the shortest path from the current switch to the destination. If this link is faulty, the following actions are performed:

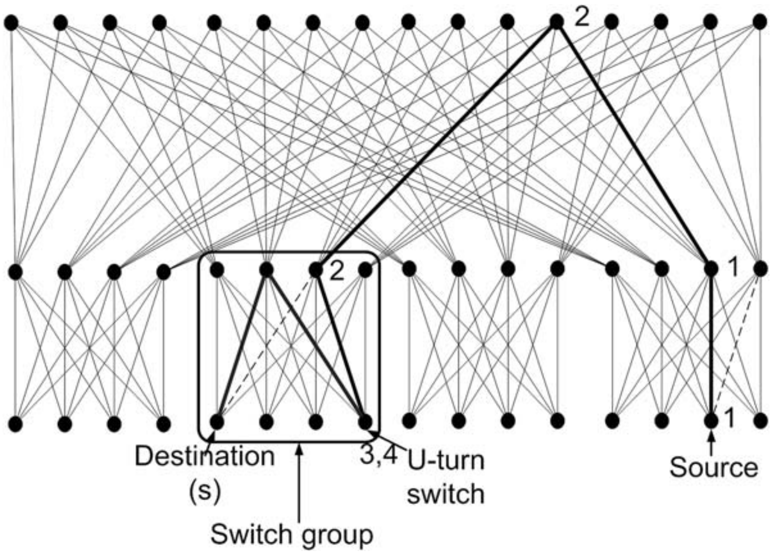


Figure 14.4 A fat tree that consists of radix 8 switches with two link faults, the processing nodes are not displayed. The faulty links are marked by dotted lines. The bold line shows the path of a packet from its source to its destination. Note how the packet is misrouted within the group via the U-turn switch. The numbers refer to the corresponding steps of the routing algorithm.

a. If the packet came from the switch tier above (from l to $l+1$), another arbitrary downward link is selected by the rerouting mechanism, which forces the packet to be rerouted. This will result in the packet being forwarded as long as there are fewer than k link faults.
b. If the packet came from the switch tier below (from $l+1$ to l), the packet is rerouted back down to the same lower-tier $(l+1)$ switch again. This is necessary to avoid livelock.

3. A switch that receives a packet from a link connected to an upper tier for which it has no downward path is a U-turn switch.
4. The U-turn switch chooses a different upward port through which to forward the packet.
5. If all upward ports from the U-turn switch have been tested, the path is disconnected and the packet must be discarded.

Which output port is chosen in the U-turn switch in point 4 and how the switch knows that all upward ports have been tested in point 5 is specific to the deterministic and adaptive routing schemes. With deterministic routing we need a strict ordering of the sequence of upward ports to test, whereas with adaptive routing it is sufficient to guarantee that that each upward port is tested at most once.

We refer to [15] for proofs that the method is actually capable of tolerating $(radix/2) - 1$ link faults in a deadlock-free manner, both for adaptive and deterministic routing.

Figure 14.5 shows the throughput properties of the rerouting method with adaptive (ADLR) and deterministic (DDLR) routing compared to a simple reconfiguration mechanism with deterministic routing (DDRC). The network has a width of 16 switches and a height of three switch tiers, not counting processing nodes, a 4-ary 3-tree. The vertical line in the plot marks the transition from the guaranteed fault tolerance region, which goes from zero up to and including $(radix/2) - 1 = 3$ link faults; beyond this limit there is only a probability of tolerating faults (i.e., from four link faults and upwards). We see that under saturated conditions, the throughput falls steadily for all methods but does so fastest for deterministic rerouting. The same degradation can be seen with ADLR as with DDRC, but it is overall slightly less efficient than DDRC. In addition, it is worth noting that DDRC loses 20,000 times as many packets as the dynamic rerouting methods because of the long reconfiguration time. For a less saturated network the throughput is largely unaffected until a large number of faults occurs.

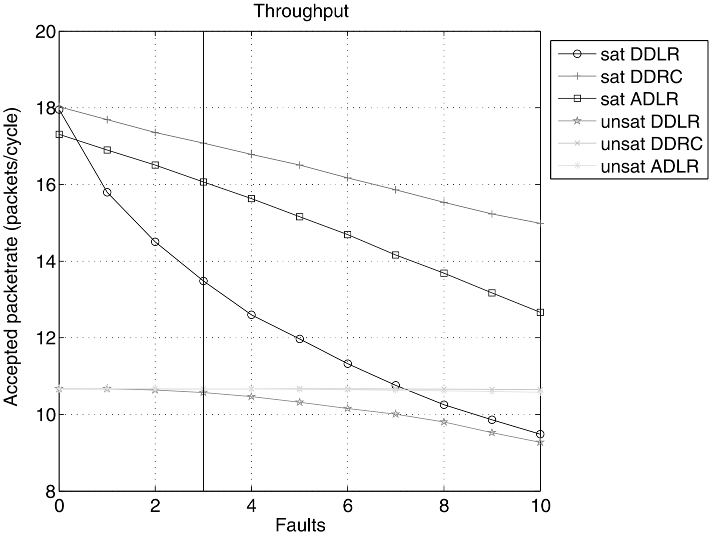


Figure 14.5 Throughput in presence of link faults, both under a saturated and unsaturated condition The vertical line marks the transition from guaranteed fault tolerance on the left side to probable fault tolerance on the right side. (For the colour version, see figure C.1 on page 643.)

14.4 Dynamic Reconfiguration

Until quite recently the prevailing view on the routing of interconnecton networks was that a network would have a static topology and that the associated routing function would not be subject to change. Two developments are now changing this in a profound way. First, the sizes of systems built these days–which comply with a static model, assuming that all components in the network are working–will impose an unacceptably short mean time between failures for the entire system. Therefore alternative routing functions that route around faulty components must be an inherent part of the design. Second, many large computing servers are divided into multiple virtual servers in their daily production situation. In order to control the sharing of interconnection resources between virtual servers, it is beneficial to be able to change the routing function so that each server becomes routing contained. By this we mean that they do not share physical resources in the network.

Changing routing functions in an interconnection network is, however, not trivial. Even if both the old and new routing functions are free from deadlocks, interactions between the old and new routing functions may lead to reconfiguration-induced deadlocks.

Early techniques typically handle this situation through static reconfiguration–meaning that application traffic is stopped and, usually, dropped from the network during the reconfiguration process (see, e.g., [30, 31]). While this approach guarantees the prevention of reconfiguration-induced deadlock, it can lead to unacceptable packet latencies and dropping frequencies for many applications, particularly real-time and quality-of-service (QoS) applications [29].

With dynamic reconfiguration, the idea is to allow user traffic to continue uninterrupted during the time that the network is reconfigured, thus reducing the number of packets that miss their real-time/QoS deadline. Earlier on, some key efforts were made towards addressing the issue of deadlock-free dynamic reconfiguration within the context of link-level flow-controlled interconnection networks. In [28], a *partial progressive reconfiguration* (PPR) technique is proposed that allows arbitrary networks to migrate between two instantiations of up*/down* routing. The effect of load and network size on PPR performance is evaluated in [55]. Another approach is the NetRec scheme [27], which requires every switch to maintain information about switches some number of hops away. Yet another approach is the *double scheme* [26], where the idea is to use two required sets of VCs in the network which act as two disjoint virtual network layers during reconfiguration. The basic idea is to first drain one virtual network layer and reconfigure it while the other is fully up and running, then to drain and reconfigure the other virtual network layer while the first is up and running, thus allowing "always on" packet delivery during reconfiguration. An orthogonal approach which may be applicable on top of all of the above techniques is described in [65], where it is shown that for UD routing, only parts of the network (i.e., the "skyline") need to be reconfigured on a network change. In [23] a methodology for developing dynamic network reconfiguration processes between any pair of routing functions is described.

All the approaches mentioned previously suffer from different shortcomings. The PPR technique [28] will only work between two routing functions that adhere to the

UD scheme. NetRec [27] is specially tailored to reroute messages around a faulty node. It basically provides a protocol for generating a tree that connects all the nodes that are neighbours to a fault and drops packets to avoid deadlocks in the reconfiguration phase. The double scheme is the most flexible, in that it can handle any topology and make a transition between any pair of deadlock-free routing functions. On the other hand, it requires the presence of two sets of VCs.

In this section we present a simple and powerful method for dynamic network reconfiguration. In contrast to the approaches mentioned previously, our method is able to handle any topology and any pair of routing functions, regardless of the number of VCs available in the network. It is directly applicable when the new routing function is available and does not require a new reconfiguration method to be derived before it can be applied. Our technique guarantees in-order delivery of packets during reconfiguration and for that reason can off-load much of the fault-handling burden from the higher-level protocols.

An Efficient Network Reconfiguration Method

We assume familiarity with the standard notation and definitions of cut-through switching. In particular, we assume that the basic notions of deadlock freedom in general and channel dependency graphs in particular are known. Readers who are unfamiliar with these notions are referred to the section on basic concepts in 14.1 and to [22].

Our focus is on the transition from one routing function to another. We will denote these two routing functions as R_{old} and R_{new}, respectively, with the subscripts taking the obvious meaning. In what follows we simply assume that each of these is deadlock free and has a cycle-free channel dependency graph, unless explicitly stated otherwise. Furthermore, we assume that if R_{old} supplies any faulty channels, the packets destined for these channels are dropped rather than stalled and that R_{new} supplies channels such that the faulty components are circumvented.

As we consider transitions from one routing function to another, channel dependency graphs are not a sufficient tool for detecting freedom from deadlocks. Even if the prevailing routing function at any given time supplies channels in a deadlock-free manner during reconfiguration, there may be configurations of packets that are deadlocked. This is because a packet may have made previous routing decisions based on old routing choices that are no longer allowed in the current routing function and by doing so has ended up in a situation where it keeps a channel dependency from a previous routing function alive. Such dependencies are called ghost dependencies [21]. We therefore need a notion of deadlocks that encompasses more information than just channel dependencies. We use a simplified form of definition 10 in [24]:

Definition 1. A set of packets is *deadlocked* if every packet in the set must wait for some other packet in the set to proceed before it can proceed itself.

We shall use this definition to show that our reconfiguration method will allow no deadlocks to form.

In the following we describe the fundamentals of our simple reconfiguration algorithm. In the description we shall assume for simplicity that there will be only a single reconfiguration process active at a time and that this reconfiguration process will complete before the next one is started.

Our method is based on two pillars: The first is that we let every packet be routed either solely according to R_{old} or solely according to R_{new}. The packets that we route solely according to R_{old} will be called *old* packets and the packets that are routed solely according to R_{new} are called *new* packets. It is very likely that a channel will be used by both old packets and new packets, so ghost channel dependencies can form from the interaction between old and new packets.

The second pillar is the following lemma:

Lemma 1. *Assume a transition from R_{old} to R_{new} in which every packet is routed solely according to R_{old} or solely according to R_{new}. Any deadlocked set of packets in such a transition will have to contain old packets waiting behind new packets.*

Proof. The proof is by contradiction. Assume a deadlocked set of packets in which no old packets wait behind new packets.

Case 1: There are no old packets in the set. In this case the set must contain only new packets that should be able to reach their destination using R_{new}. This implies that R_{new} is not deadlock free and we have a contradiction.

Case 2: There are old packets in the set. Since we assume that no old packet waits behind new packets, the old packets must all be waiting behind each other. In that case there must exist a deadlocked set containing only old packets. This implies that R_{old} is not deadlock free and we have a contradiction.

A consequence of this lemma is that if we make sure that packets routed according to R_{old} will never have to wait behind packets routed according to R_{new}, we will have achieved freedom from deadlock even if R_{new} packets wait behind R_{old} packets. This can be achieved by letting all channels transmit a token that indicates that all packets injected before this token shall be routed according to R_{old} and all packets injected after this token shall be routed according to R_{new}. We let this token flood the network in the order of the channel dependency graph of R_{old} and for each channel it traverses, it means the same thing: All packets transmitted across the channel before this token shall be routed according to R_{old} and all packets after this token shall be routed according to R_{new}. Every packet routed according to R_{new} will simply have to wait for the token to have passed before it enters a channel. That way no packet routed according to R_{old} will ever have to wait for a packet routed according to R_{new} to proceed, thus avoiding deadlock. A more formal description of one version of the process follows.

1. Let each injection link send a token onto all of its channels indicating that no packets that have been routed according to R_{old} will arrive on this channel.
2. Let each switch do the following.
   - For each input channel do the following:
     a. Continue using R_{old} until a token has arrived at the head of the queue[4].

[4] We make the usual assumption that a packet is routed only when it is at the head of the input queue.

b. When the token has made it to the head of the queue, change into R_{new} for this input channel.
c. Thereafter forward packets only to those output channels that have transmitted the token.

- For each output channel do the following:
  a. Wait until all input channels from which the output channel can expect to receive traffic according to R_{old} have processed the token.
  b. Thereafter transmit a token on the output channel.

The following results can now be derived for this process.

Observation 1 *All input channels on all switches use* R_{old} *until they process the token and, thereafter, use* R_{new}.

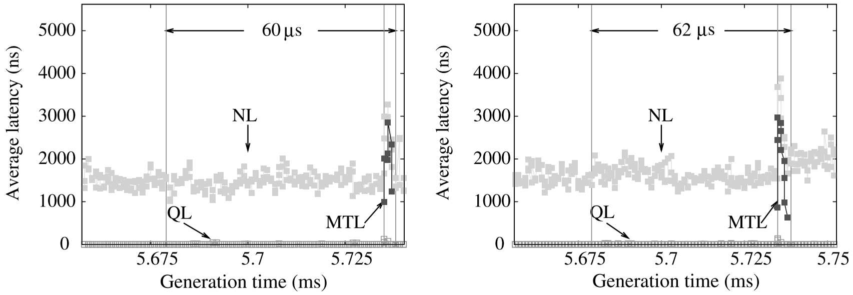
Lemma 2. *The process terminates with all channels having transmitted the token if* R_{old} *has a cycle-free dependency graph.*

Lemma 3. *If both* R_{old} *and* R_{new} *are deterministic, in-order packet delivery is maintained during the reconfiguration process.*

For proofs of the above results we refer to [1].

Evaluation

The format of this bookchapter does not allow for a full evaluation of the reconfiguration procedure. Here we restrict ourselves to presenting two plots, showing the latency of the different packets in the network while reconfiguration is taking place.



(a) Medium Traffic Intensity

(b) High Traffic Intensity

Figure 14.6 Average packet latency at various generation times for uniform traffic, where various components of latency are broken down as follows: QL is Queue Latency, NL is Network Latency, and MTL is Maximum Token Latency. The vertical bars represent the start and the end of the reconfiguration respectively. (For the colour version, see figure C.2 on page 644.)

The experiments were conducted in an 8x8 torus, where a fault is injected, new routing tables are uploaded, and reconfiguration is performed. Figure 14.6 shows average packet latency plotted against generation time under medium and high traffic load rates. The start and end times for reconfiguration are indicated with vertical lines. An extra vertical line is added that indicates when the routing tables have been uploaded and the reconfiguration tokens are being injected. As can be seen in the plots, the method allows for uninterrupted traffic flow while reconfiguration takes place and thus the method's main goal of allowing true dynamic change of routing algorithms is reached. There is, however, a temporary increase in packet latency while the tokens traverse the network. For a more comprehensive analysis of the performance of this method and a full discussion of its performance we refer to [1].

14.5 Technology Transfer and Collaboration

An important asset is that ICON has been able to establish collaboration with the following prominent industrial partners: Sun Microsystems, Inc., Dolphin Interconnect Solutions ASA, Silicon Graphics, Inc., and Mellanox Technologies, Ltd. These collaborations may result in the adoption of ICON technology in future products launched by these manufacturers, which again will increase the impact of our research.

It is not straightforward to establish collaboration with the industry: Besides having a competence they are interested in, there are challenges related to intellectual property rights and obtaining funding support. We strongly believe that the way Simula is organised and its collaboration philosophy have made this process smoother compared to if ICON had been located at a university. Simula and ICON have advocated flexibility and a win-win attitude when negotiating collaboration with the industry. Below we say a bit more about the collaboration ICON has established.

Industrial Take-up of LASH

The OpenFabrics Alliance. In light of the fact that LASH fits nicely to InfiniBand, in 2006 ICON contacted OpenFabrics to check for their interest in LASH. As a result a dialogue was established with the community and the following year ICON implemented LASH in the OpenFabrics software framework.

Silicon Graphics. Following the availability of LASH in OpenFabrics, ICON was contacted by the HPC company Silicon Graphics (SGI). This company wanted to use LASH for their next supercomputer to be networked as a torus topology. A dialogue between SGI and ICON was established where ICON supported them in the process of adopting LASH. As documented by the SGI Altix ICE Systems Administrators Guide [17], Silicon Graphics recommends using LASH for Altix ICE systems with more than 128 nodes. Also SGI has made an effort to optimise LASH in OpenFabrics and has released a new version thereof.

Sun Microsystems and Sandia National Laboratory. Recently ICON established a research project with Sun Microsystems, where the overall objective is to advance HPC networking technology based on InfiniBand (we will come back to this project below). Sun Microsystems also become aware of the availability of LASH in Open-

Fabrics and one of the first deliverables of this project was to assess LASH (in the context of OpenFabrics) with respect to a large switch fabric to be laid out as a three-dimensional torus network and determine where this supercomputer is to be installed at Sandia National Laboratory. This study concluded positively about the LASH hypothesis and at Super Computing 2008 Sandia National Laboratory announced that they will use LASH in their next generation of supercomputers [16].

Sun Microsystems

Sun Microsystems has a branch in Norway that develops switch components and large switch fabrics, including management software based on the InfiniBand technology. For some years Sun Microsystems and ICON have discussed challenges in interconnection networking and in this context have published joint papers.

Based on this established relationship, both parties saw mutual benefits in a formalisation of this cooperation. This has led to a project aimed at combining academic research and industry relevance that will benefit both parties as well as the HPC community in general. It is a jointly funded project that will undertake fundamental research in interconnection networks and HPC. The project goals are to meet the network technology requirements for the next generation of supercomputers. The agreement was announced in October 2008 for an initial three-year life span. The project will be hosted on-site at Simula and manned by ICON researchers (one post-doctoral fellow and two doctoral candidates), who will cooperate closely with Sun Microsystems in Norway.

Sun Datacenter 3456 Switch. This collaboration has already resulted in a technology transfer: The Sun Datacenter 3456 constellation switch is using a routing algorithm developed by ICON on its internal fat tree structure[5]. For example, the Texas Ranger supercomputer at the Texas Advanced Computing Center (TACC) is considering the use of this technology. The Texas Ranger supercomputer is at the time of this writing the sixth most powerful machine in the world.

Dolphin Interconnect Solutions

Dolphin Interconnect Solutions is an HPC company (headquartered in Norway) developing products used to connect multiple computers together to serve demanding applications. Their solutions include both software and hardware architecture for creating powerful cluster systems.

For some years ICON has been collaborating with Dolphin where the target was to develop a dynamic fault tolerance solution for their scalable-coherence-interface-based cluster computers. In 2006–2007 the solution was implemented successfully both in firmware and hardware. It has proved to reduce the time span for handling failover from around ten seconds to a few milliseconds, making their system better suited to serve, for example, MySQL® and Oracle applications.

[5] This method is yet to be published, but its current label is “view-based routing.”

14.6 Future Outlook

The computer architecture and HPC communities, herein interconnection networks, continuously evolve and are faced with many severe research challenges in the future. The ICON project will concentrate on the following three problem areas over the next five years:

- To further develop solutions for generic interconnection networking, encompassing routing, fault-tolerance, QoS, and congestion management.
- To develop solutions for network on chip (NoC/multicore) systems, encompassing flexible and power-efficient routing and job allocation and exploring the latency-power trade-off in the design of NoC topologies.
- To develop solutions for utility computing data centres (UCDCs), encompassing flexible partitioning (virtualisation), faulttolerance, and predictable service.

Let us try to motivate a bit more the focus of those areas where ICON will contribute in the years to come. High performance computing is steadily becoming an increasingly important industry, simply because modern society's need for computing power is continuously increasing. Existing applications tend to become more demanding in order to produce more accurate and reliable results and to serve larger application instances. New application areas are forthcoming, as, for instance, the automotive industry, which has developed simulation models to be used for evaluating the safety properties of newly developed cars, instead of using real ones, an option that has become too costly today. These simulations demands a lot of computing power and the automotive industry will have to undertake HPC investments. Further, considering the vast amount of data produced today, exemplified through home digital still and video cameras and the need to store, process, protect, and thus move these data, there is a clear market need for affordable storage and processing systems. The tendency is that in the future the HPC community will rely more and more on cluster-based computers that deploy generic interconnection networks. Another related trend is that the community goes in the direction of using more open-source. This is being confirmed by the increased popularity of the OpenFabrics Alliance, which aims to develop and promote InfiniBand protocol stack software and management functionality.

Recently ICON has taken two new strategic manoeuvres in order to increase the impact of our HPC research: (1) ICON has invested in an InfiniBand cluster. This will enable us to implement proof-of-concept prototypes of our solutions and write more comprehensive research papers. It will also make it easier for us to contribute our solutions to the OpenFabrics community. (2) ICON has established a long-term collaboration research project with Sun Microsystems to conduct fundamental research in interconnection networks and high performance computing. These steps should put ICON in a good position to be recognised as a major and regular contributor to the design of solutions for generic interconnection networks that can also be adopted by the industry in their future products.

The indications that Moore's law will soon meet its end will make multicore system and NoC architectures crucial to meet the future challenges of designing high-performance CPUs. The number of cores is foreseen to increase significantly in a five to ten year horizon, making the NoC a fundamental component of the CPU. This leads to several challenges, for instance, the memory bandwidth requirements of massively multicore chips present a tremendous challenge to designers. A high-speed, high-bandwidth on-chip network is required; in many cores, a shared bus is no longer sufficient. Instead, direct networks, as, for example, the two-dimensional mesh topology, are being proposed. For some chips, such as those with heterogeneous cores, a less regular topology may be used. Even manufacturing defects or run-time failures might convert the topology into an irregular one, requiring routing algorithms that can handle such irregularities. Power management is another issue that will become important in the future in order to offer energy-aware or green computing. A typical situation in multicore chips might be that not all the cores will be in use simultaneously and those cores, and associated network components, should then be turned off to save power. This may require flexible power-aware routing that can route around the turned-off components and still provide connectivity. Moreover, with respect to multicore chips, three key metrics reveal themselves to be of major importance: performance, fault tolerance (including yield), and power consumption. A solution that optimises all three of these metrics is challenging. As the number of cores increases, the importance of the NoC also grows and chip designers should aim to optimise these three key metrics in the NoC context as well. Together with the Technical University of Valencia, ICON has recently published a position paper that identifies and discusses the main properties that a NoC must exhibit in order to enable such optimisations. In particular, use of virtualisation techniques at the NoC level are of crucial importance [14].

It has long been realised that computing power has the potential of being provided as a utility similarly to how electricity is an available utility [20]. Until recently, however, the limited data transfer capacity of communication networks has hindered such a development. Thus, companies in need of computing power have had no choice but to purchase and maintain their own private data centres. Such in-house provision of computing power is costly, complex, and may deflect focus from a company's core activities. In addition, considering the increased awareness about global warming caused by CO_2 emission, the immense electricity consumption, and, in general, low utilisation ratios of the high number of current private data centres raise environmental and ethical concerns. The recent wide upgrade of communication networks into using optical fiber links has significantly increased the capacity of communication networks and made utility computing possible. Vendors have provided various solutions to utility computing, such as Sun's N1 [25] and HP's Adaptive Enterprise [5]. Recent examples of utility computing services now being offered include Google [13], Sun Grid Compute Utility [6], and Amazon Elastic Compute Cloud [19].

We view a utility computing data centre (UCDC) as a large collection of resources in which each resource falls into one of the following categories: compute nodes (CPUs with memory), storage nodes (disks or tapes), and access nodes (gateways to external networks). An interconnection network links these resources. The UCDC assigns resources on demand to customers, who request a subset of the resources in the data centre for a defined period. Typically, customers pay only for the resources they require and only for the period they require them. As UCDCs become widespread, new user groups will get access to computing power due to the expected relatively low cost. The jobs or services running in a UCDC typically have diverse characteristics, such as different resource requirements, running times, software quality, security requirements, and importance. In addition to the classic HPC applications, a job might be, for example, an ad hoc web service set up to host a sports event for a limited period of time. Furthermore, when creativity is no longer restrained by the cost of computing power, applications may appear that we do not foresee today. In research on interconnection networks, the common assumption has been that the system is executing only one job or class of jobs concurrently and that the interconnection network's task is to maximise overall performance. A UCDC's diverse requirements therefore generate a set of problems that researchers have only marginally studied before. Recently ICON published a position paper which identifies architectural challenges that a UCDC poses on the interconnection network [18].

14.7 Final Remarks and Conclusions

The work on interconnection networks at Simula started at a time when the field was not considered fashionable in large parts of the research community. This generally made it hard to build and sustain research projects of some size, even if the extrapolation of technology trends indicated the prospects of the field were very good. At Simula there is a policy to make strategic decisions on selected areas of research and to construct long-term focussed research projects to pursue these selected research goals. This policy enabled the buildup of a focussed research team in this area. This strategy has payed off handsomely. When industrial interest in the field came back as a result of the interconnection network again becoming a limiting factor of scalability and performance, this group was well positioned.

The group has gone through all stages from basic research on network structures that was not even planned by industry at the time to substantial collaborative projects with industrial development groups. The collaboration with development teams will continue in the following years, particularly in the area of architectures for utility computing. In on-chip networks, however, we see a new area for more basic research. It is our hope that we will be able to take this new area forward towards application in the years to come.

Acknowledgements

We would like to thank José Miguel Montañana for kindly providing the performance figures of the dynamic reconfiguration method presented in section 14.4.

References

[1] O. Lysne, J. M. Montanana, J. Flich, J. Duato, T. M. Pinkston, and T. Skeie. An efficient and deadlock-free network reconfiguration protocol. *IEEE Transactions of Computers*, 57(6):762–779, 2008.

[2] W. J. Dally and B. Towles. *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.

[3] J. Duato. A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks. *IEEE Trans. Parallel Distrib. Syst.*, 6(10):1055–1067, 1995.

[4] R. Seifert. *Gigabit Ethernet*. Addison Wesley Pub Co., 1998.

[5] Adaptive Enterprise: Business and IT Synchronized to Capitalize on Change. White paper, HP, 2005.

[6] S. Microsystems. *Sun Grid Compute Utility – Reference Guide*. Part no. 819-5131-10, 2006.

[7] F. Petrini, W. chun Feng, A. Hoisie, S. Coll, and E. Frachtenberg. The Quadrics network: High-performance clustering technology. *IEEE Micro*, 22(1):46–57, /2002.

[8] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su. Myrinet: A gigabit-per-second local area network. *IEEE Micro*, 15(1):29–36, 1995.

[9] Top 500 supercomputer sites. http://www.top500.org/, October 2008.

[10] F. Petrini and M. Vanneschi. K-ary N-trees: High performance networks for massively parallel architectures. Technical Report TR-95-18, 15, 1995.

[11] X.-Y. Lin, Y.-C. Chung, and T.-Y. Huang. A multiple lid routing scheme for fat-tree-based infiniband networks. *IPDPS*. IEEE Computer Society, 2004.

[12] *18th International Parallel and Distributed Processing Symposium (IPDPS 2004), CD-ROM / Abstracts Proceedings, 26-30 April 2004, Santa Fe, New Mexico, USA*. IEEE Computer Society, 2004.

[13] Google. www.google.com.

[14] J. Flich, J. Duato, T. Sødring, Å. G. Solheim, T. Skeie, O. Lysne, and S. Rodrigo. On the potential of noc virtualization for multicore chips. *Proc. International Workshop on Multi-Core Computing Systems (MuCoCoS'08)*. IEEE Computer Society, 2008.

[15] F. Sem-Jacobsen. Towards a unified interconnect architecture: Combining dynamic fault tolerance with quality of service, community separation, and power saving. *PhD thesis*. University of Oslo, 2008.

[16] Sun. Red sky at night, sandia's delight.

[17] S. Graphics. *SGI Altix ICE System Administrator's Guide*. Silicon Graphics, version 001 edition, 2008.

[18] O. Lysne, S.-A. Reinemo, T. Skeie, Å. G. Solheim, T. Sødring, L. P. Huse, and B. D. Johnsen. The interconnection network – architectural challenges for utility computing data centres. *IEEE Computer*, 41(9):62–69, 2008.

[19] Amazon.com. Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/.

[20] N. Carr. *The Big Switch*. W. W. Norton, New York, London, 2008.

[21] J. Duato, O. Lysne, R. Pang, and T. M. Pinkston. Part I: A theory for deadlock-free dynamic network reconfiguration: A theory for deadlock-free dynamic network reconfiguration. *IEEE Transactions on Parallel Distributed Systems,* 16(5):412–427, 2005.

[22] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: An Engineering Approach.* Morgan Kaufmann Publishers, 2003.

[23] O. Lysne, T. M. Pinkston, and J. Duato. Part ii: A methodology for developing deadlock-free dynamic network reconfiguration processes. *IEEE Transactions on Parallel Distributed Systems,* 16(5):428–443, 2005.

[24] S. Warnakulasuriya and T. M. Pinkston. A formal model of message blocking and deadlock resolution in interconnection networks. *IEEE Transactions on Parallel and Distributed Systems,* 11(3):212–229, 2000.

[25] N1 Grid Engine 6 features and capabilities. Sun Microsystems White Paper, 2004.

[26] T. Pinkston, R. Pang, and J. Duato. Deadlock-free dynamic reconfiguration schemes for increased network dependeability. *IEEE Transactions on Parallel and Distributed Systems,* 14(8):780–794, Aug. 2003.

[27] N. Natchev, D. Avresky, and V. Shurbanov. Dynamic reconfiguration in high-speed computer clusters. *Proceedings of the International Conference on Cluster Computing,* pages 380–387, Los Alamitos, 2001. IEEE Computer Society.

[28] R. Casado, A. Bermúdez, J. Duato, F. J. Quiles, and J. L. Sánchez. A protocol for deadlock-free dynamic reconfiguration in high-speed local area networks. *IEEE Transactions on Parallel and Distributed Systems,* 12(2):115–132, Feb. 2001.

[29] J. Fernández, J. García, and J. Duato. A new approach to provide real-time services on high-speed local area networks. *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS-01),* pages 124–124, Los Alamitos, CA, Apr. 23–27 2001. IEEE Computer Society.

[30] T. L. Rodeheffer and M. D. Schroeder. Automatic reconfiguration in Autonet. *Proceedings of 13th ACM Symposium on Operating Systems Principles,* pages 183–197. Association for Computing Machinery SIGOPS, Oct. 1991.

[31] D. Teodosiu, J. Baxter, K. Govil, J. Chapin, M. Rosenblum, and M. Horowitz. Hardware fault containment in scalable shared-memory multiprocessors. *Proceedings of the 24th Annual International Symposium on Computer Architecture (ISCA-97),* volume 25, 2 of *Computer Architecture News,* pages 73–84, New York, 1997. ACM Press.

[32] A. Kvalbein, A. F. Hansen, T. Čičić, S. Gjessing, and O. Lysne. Multiple routing configurations for fast IP network recovery. *IEEE/ACM Transactions on Networking,* 2009 (To Appear).

[33] I. T. Theiss and O. Lysne. Froots, a fault tolerant and topology agnostic routing technique. *IEEE Transactions on Parallel and Distributed Systems,* 17(10):1136–1150, 2006.

[34] N. Sharma. Fault-tolerance of a min using hybrid redundancy. *Simulation Symposium, 1994., 27th Annual,* pages 142–149, Apr 1994.

[35] F. O. Sem-Jacobsen, T. Skeie, O. Lysne, and J. Duato. Dynamic fault tolerance with misrouting in fat trees. *Proceedings of the International Conference on Parallel Processing (ICPP),* pages 33–45. IEEE Computer Society, 2006.

[36] N. A. Nordbotten and T. Skeie. A routing methodology for dynamic fault tolerance in meshes and tori. *International Conference on High Performance Computing (HiPC),* LNCS 4873, pages 514–527. Springer-Verlag, 2007.

[37] C. Carrion, R. Beivide, J. A. Gregorio, and F. Vallejo. A flow control mechanism to avoid message deadlock in k-ary n-cube networks. *High-Performance Computing, International Conference on,* 0:322, 1997.

[38] M. E. Gómez, N. A. Nordbotten, J. Flich, P. López, A. Robles, J. Duato, T. Skeie, and O. Lysne. A routing methodology for achieving fault tolerance in direct networks. *IEEE Transactions on Computers,* 55(4):400–415, 2006.

[39] J.-D. Shih. Fault-tolerant wormhole routing in torus networks with overlapped block faults. *Computers and Digital Techniques, IEE Proceedings –,* 150(1):29–37, 2003.

[40] J.-D. Shih. A fault-tolerant wormhole routing scheme for torus networks with nonconvex faults. *Information Processing Letters,* 88(6):271–278, 2003.

[41] S. Chalasani and R. Boppana. Fault-tolerant wormhole routing in tori. *Proceedings ACM International Conference on Supercomputing,* pages 146–155, 1994.

[42] S. Park, J.-H. Youn, and B. Bose. Fault-tolerant wormhole routing algorithms in meshes in the presence of concave faults. *Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International,* pages 633–638, 2000.

[43] H. Gu, Z. Liu, G. Kang, and H. Shen. A new routing method to tolerate both convex and concave faulty regions in mesh/tori networks. *Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT. Sixth International Conference on,* pages 714–719, Dec. 2005.

[44] S. Kim and T. Han. Fault-tolerant wormhole routing in mesh with overlapped solid fault regions. *Parallel Computing,* 23:1937–1962, 1997.

[45] P. Sui and S. Wang. An improved algorithm for fault-tolerant wormhole routing in meshes. *IEEE Transactions on Computers,* 46(9):1040–1042, 1997.

[46] C. Cunningham and D. Avresky. Fault-tolerant adaptive routing for two dimensional meshes. *Proceedings Symp. on High-Performance Computer Architecture,* pages 122–131, 1995.

[47] Openfabrics Alliance. www.openfabrics.org.

[48] Å. Solheim, O. Lysne, T. Skeie, T. Sødring, and I. Theiss. Routing for the asi fabric manager. *IEEE Communication Magazine,* 44(7):39–44, 2006.

[49] O. Lysne, T. Skeie, S.-A. Reinemo, and I. Theiss. Layered routing in irregular networks. *IEEE Transactions on Parallel and Distributed Systems,* 17(1):51–65, 2006.

[50] A. Mejía, J. Flich, S.-A. Reinemo, and T. Skeie. Segment-based routing: An efficient fault-tolerant routing algorithm for meshes and tori. *Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium,* pages 1–10, 2006.

[51] A. Joraku, K. Koibuchi, and H. Amano. An effective design of deadlock-free routing algorithms based on 2d turn model for irregular networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(3):320–333, March 2007.

[52] J. Sancho, A. Robles, and J. Duato. An effective methodology to improve the performance of the up*/down* routing algorithm. *IEEE Transactions on Parallel and Distributed Systems*, 15(8):740–754, 2004.

[53] H. Sullivan and T. Bashkow. A large scale, homogeneous, fully distributed parallel machine. *Proceedings of the 4th International Symposium on Computer Architecture*, March 1977.

[54] R. V. Boppana and S. Chalasani. Fault-tolerant wormhole routing algorithms for mesh networks. *IEEE Transactions on Computers*, 44(7):848–864, 1995.

[55] R. Casado, A. Bermúdez, F. J. Quiles, J. L. Sánches, and J. Duato. Performance evaluation of dynamic reconfiguration in high-speed local area networks. *Proceedings of the Sixth International Symposium on High-Performance Computer Architecture*, 2000.

[56] S. Chalasani and R. V. Boppana. Communication in multicomputers with nonconvex faults. *IEEE Transactions on Computers*, 46(5):616–622, 1997.

[57] L. Cherkasova, V. Kotov, and T. Rockicki. Fibre channel fabrics: Evaluation and design. *29th Hawaii international conference on system sciences*, 1995.

[58] A. A. Chien and J. H. Kim. Planar-adaptive routing: Low-cost adaptive networks for multiprocessors. *Journal of the Association for Computing Machinery*, 42(1):91–123, 1995.

[59] W. J. Dally and C. L. Seitz. The torus routing chip. *Distributed Computing*, 1:187–196, 1986.

[60] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Transactions on Computers*, C-36(5):547–553, 1987.

[61] C. J. Glass and L. M. Ni. The turn model for adaptive routing. *Journal of the Association for Computing Machinery*, 41(5):874–902, 1994.

[62] C. J. Glass and L. M. Ni. Fault-tolerant wormhole routing in meshes without virtual channels. *IEEE Transactions on Parallel and Distributed Systems*, 7(6):620–636, June 1996.

[63] I. T. Association. Infiniband architecture specification.

[64] D. H. Linder and J. C. Harden. An adaptive and fault tolerant wormhole routing strategy for *k*-ary *n*-cubes. *IEEE Transactions on Computers*, 40(1):2–12, 1991.

[65] O. Lysne and J. Duato. Fast dynamic reconfiguration in irregular networks. *Proceedings of the 2000' International Conference of Parallel Processing, Toronto (Canada)*, pages 449–458. IEEE Computer Society, 2000.

[66] O. Lysne and T. Skeie. Load balancing of irregular system area networks through multiple roots. *Proceedings of 2nd International Conference on Communications in Computing*, pages 142–149, 2001.

[67] L. M. Ni and P. McKinley. A survey of wormhole routing techniques in direct networks. *Computer*, 26:62–76, 1993.

[68] W. Qiao and L. M. Ni. Adaptive routing in irregular networks using cut-through switches. *Proceedings of the 1996 International Conference on Parallel Processing (ICPP '96),* pages 52–60. IEEE Computer Society, 1996.

[69] J. C. Sancho, A. Robles, and J. Duato. A new methodology to compute deadlock-free routing tables for irregular networks. *Proceedings of the Workshop on Communication, Architecture, and Applications for Network-Based Parallel Computing (CANPC'00),* 2000.

[70] M. D. S. et.al. Autonet: a high-speed, self-configuring local area network using point-to-point links. SRC Research Report 59, Digital Equipment Corporation, 1990.

[71] T. Skeie, O. Lysne, and I. Theiss. Layered shortest path (LASH) routing in irregular system area networks. *In proceedings of Communication Architecture for Clusters,* 2002.

15

PROVIDING RESILIENCE IN COMMUNICATIONS NETWORKS

Amund Kvalbein and Yan Zhang

Amund Kvalbein · Yan Zhang
Simula Research Laboratory

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_15,

PROJECT OVERVIEW

Resilient Networks

An increasing number of services that people and businesses rely on in their everyday activities runs over the Internet. Failures in the network infrastructure can disrupt access to important functions such as telephony, banking, trading systems, and a wide range of online resources, just to mention a few examples. With our increased dependence on computer networks, their resilience against failures becomes an obvious concern. Protecting the user experience against failures requires a range of different resilience mechanisms throughout the network, from the wireless hop that connects a mobile user to a base station, to the core infrastructure that binds the Internet together on a large scale. In accordance with Simula's strong focus on performing research with a practical relevance for the industry, network resilience was identified as a key research area. This led to the establishment of the Resilient Networks project, financed over a five year period by the Ministry of Transport and Communications.

Scientific Challenges

A main focus of network resilience research at Simula is on mechanisms that can maintain an acceptable level of service when faced with various challenges to normal operations. In the core infrastructure of a network, these challenges can be component failures, attacks, or unexpected changes in the input traffic patterns. In the wireless domain, special challenges are connected to the unreliability of the wireless channel, mobility, interference, and power scarcity. We approach these challenges through a combination of mathematical modelling, simulations, measurements, and prototype implementation.

Obtained and Expected Results

Our main scientific result to date is a method for fast recovery from component failures in the network infrastructure. By planning alternative routes through the network in advance, our solution allows data traffic to continue uninterrupted even when a link or a router breaks down. This is particularly important for applications with strong demands for continuous operations, such as trading systems and process control systems. Our method has been widely disseminated through leading scientific publication channels. The method has also been patented, and a spin-off company (Resiliens A/S) was set up in 2008 with the goal of integrating our solution in a commercial product.

PROVIDING RESILIENCE IN COMMUNICATIONS NETWORKS

15.1 Introduction

Modern society relies heavily on communications networks. The ability to communicate, access information, and control processes through networks has revolutionized the way people and businesses organize their operations. Reliability and availability of these networks are therefore central concerns, and the consequences of their failures are potentially disastrous.

Two important trends in the context of network reliability are *convergence in the networking layer* and *diversification in access technologies*. On the one hand, there is a trend that more and more applications are transported over networks using the internet protocol (IP). Applications such as television, and fixed and mobile telephony, which traditionally have been transmitted using distinct technologies in specialized networks, are now increasingly offered over a common IP network. This trend brings advantages for both consumers and network operators in terms of new functionality and reduced costs. However, the migration of new applications to IP networks increases the demand for network resilience if end users are to experience a quality and availability that is similar to what has been delivered by the specialized networks. Important challenges in this context are fast recovery when there is a failure in the network and the ability to maintain an acceptable service level under stressful conditions, such as during sudden increases or changes in the traffic distribution.

At the same time, there is a diversification in the access technology used to connect to the Internet. In particular, wireless access has experienced an explosion of development, with an increase in the number of mobile terminals (mobile phones, laptops, BlackBerries, PDAs, etc.). Corresponding to this trend, for many users, it has become a basic requirement to be able to access the Internet and enterprise networks anytime and anywhere. Differing from traditional telecommunication networks, wireless networks have limited bandwidths, unreliable wireless channels, insecure radio interfaces, and constrained power. Due to these disadvantages, wireless services are often unstable, of low quality, or even experience forced termination. Therefore, in both academia and industry, providing resilience capability in wireless networks is far from maturity.

The goal of the Resilient Networks project is to *enable more reliable networks with the capability of maintaining an acceptable level of service when faced with challenges to normal operation*. Examples of such challenges are component failures and disasters, such as the 2001 fire in a Baltimore train tunnel that disrupted much of the connectivity for Washington DC [49], power depletion, resource scarcity, legitimate but unexpected changes in traffic patterns, or even malicious attacks.

The Resilient Networks project is funded by a 5-year contract with the Norwegian Ministry of Transport and Communications running from 2006 to 2010, for a total of 27 million NOK. At the beginning of the project, the core staff consisted of one postdoctoral fellow and two PhD students, in addition to two research scientists in adjunct positions. The project has grown as its funding has increased, and today it consists of one research scientist, two postdoctoral fellows, and three PhD students, in addition to two research scientists in adjunct positions.

The Resilient Networks project collaborates extensively with our distinguished peer-groups, through exchanges of students, visits by faculty, and joint publications. Strong collaborations that have led to several joint publications have been established with groups at Georgia Institute of Technology (USA), the University of Arizona (USA), Universität Würzburg (Germany), and The University of British Columbia (Canada).

It is an important goal that the conducted research be relevant to the networking industry. The industry provides guidance in selecting problems that are perceived as real, and we seek to devise solutions that are useful within that context. A main collaborator in this respect is the large Norwegian telecom company Telenor, whom we meet with regularly for mutual benefit. Our research in fast recovery mechanisms has also led to the establishment of a spin-off company, Resiliens AS, that provides resilience solutions for the routing and switching industry.

The history and form of the Resilient Networks project has been strongly shaped by the Simula culture. When starting an activity in a new area, Simula places a heavy emphasis on selecting problems that have a clear relevance for industry. Through discussions with our industrial partners, notably Telenor, resilience–both in the core of the network and in the wireless access–was identified as an area in which basic research was needed to address distinct challenges faced by network operators. Simula's funding structure then made it possible to gradually build an expert group in this strategically important area. The requirement of industrial relevance continues to be a guiding principle in the Resilient Networks project through a focus on solutions that are compatible with existing network architectures.

At the organizational level, Simula's approach using project-based PhD supervision continues to be an important success factor for the project. This creates a group feeling that we consider important for the working environment. The directed basic research culture within Simula allows the project members freedom to focus on carefully selected, meaningful problems, within the frame set by the project goals.

Research overview

The Resilient Networks project targets several important challenges for improving the reliability, availability, scalability, and Quality-of-Service (QoS) in IP and wireless networks. We examine challenges that arise in all segments of various communications networks–close to the end user at the edge of the network, in the core of an operator's transport network, and on a global scale in the interconnection between networks. Our aim is to better understand the factors that limit network resilience and to devise solutions that address these issues. The chief scientific motivation is the provisioning of end-to-end resilience with cost-effective approaches. To achieve this, we seek solutions that are relevant to the operational networking community. By this, we mean that our solutions should be deployable by making changes to or within the existing network infrastructure, and they should not require a completely new network architecture.

In networks of a certain size, the failure of network components, such as links, routers, interface cards, and so forth will be a part of normal operations. To maintain high availability and avoid the loss of data, it is important to have mechanisms that rapidly route traffic around a failed component with minimal disruptions. Traditionally, such recovery is done by a routing protocol, for example, Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS). After a failure, the routing protocol distributes the updated topology information to all nodes, which can then re-run the routing calculations and install new routes to all destinations. This process involves all routers in the network, and typically takes a few seconds to complete. In section 15.2, we present methods that radically decrease the recovery time after component failures. Our main contribution in this field, called *Multiple Routing Configurations* (MRC), is based on maintaining a limited amount of extra routing information in the routers, so they can locally determine an alternative path when a failure occurs, without coordinating with other routers in the network. This allows recovery in a time frame of milliseconds. MRC requires making only small changes to any current routing infrastructure.

The introduction of new applications also poses new challenges in the way networks adapt to changes in the input traffic. Instead of merely providing good performance over long timescales (say hours), it is now important that an Internet service provider (ISP) also deal effectively with short-term overload conditions that may last from a few seconds to several minutes. For such timescales, it is not possible to rely on mechanisms requiring human intervention. In section 15.3, we present *Routing Homeostasis*, a routing method in which the primary goals are to be robust to both traffic load variations and topology variations, while offering good routes in terms of mitigating delay. We seek to achieve this with a method that is incrementally deployable in the current routing infrastructure and stable without imposing a high signalling overhead on the routers. Homeostasis is based on spreading traffic over multiple available paths, coupled with the ability to respond to load changes by dynamic load balancing between these paths. Our method takes its name from the biological property of homeostasis, the mechanism by which organisms manage to maintain a robust and stable function, despite large and unpredictable changes in their environment.

The most important routing protocol in the Internet is the Border Gateway Protocol (BGP), which is used for interdomain routing between independent networks. Recently, there has been significant concern in the research and operational communities about the scalability of BGP [24]. BGP scalability is an issue in two different respects: an increasing size of routing tables and an increasing rate of routing updates (often called *churn*). In section 15.4, we describe our work in this area, which focuses on the latter. Earlier work [13] reports an alarming increase in churn, but the existing explanations for this trend are largely anecdotal. A good understanding of BGP scalability in terms of churn is important in determining whether this protocol can continue to be the glue that binds the Internet together in the future or if a radically different routing architecture will be needed. Our main focus is on characterizing and understanding the growth in churn, and how it depends on 1) the events that trigger routing updates, 2) the topological characteristics of the Internet, and 3) the various BGP mechanisms used to limit update traffic. We pursue this goal through a combination of measurements, using publicly available routing traces [27, 28] and simulations.

To guarantee end-to-end resilience, mobile access is an indispensable component. Due to the fundamental difference between the Internet and wireless communications, wireless resilience shows large variation in both the architecture and protocols. Additionally, this results in different rationales for wireless resilience design, analysis, and solutions.

Wireless resilience is highly dependent on the specific kinds of wireless systems and services in operation. Since various wireless systems have different characteristics, the demands in achieving resilience are diverse; consequently, so are the aspects of problem formulation, techniques, and solutions. Wireless Sensor Networks (WSNs) is a promising technology with wide applications, for example, environment and habitat monitoring, health care applications, battlefield surveillance, and traffic control. In WSNs, sensor nodes are usually unattended, resource-constrained, and non-rechargeable. Hence, efficient utilization of the limited energy source is the most important concern in providing resilience in sensor networks. We propose an energy-efficient and QoS-aware packet forwarding scheme with the aim of minimizing energy consumption, subject to an end-to-end delay requirement. The adaptive transmission rate, energy efficiency, and routing decisions are taken into account simultaneously. The results of our efforts indicate significantly conserved energy consumption with a guaranteed delay requirement. This mechanism is particularly useful for real-time monitoring in unattended areas, one of the main applications of sensor networks.

Resilience in wireless networks is closely related to the QoS guarantee. Normally, higher resilience indicates higher QoS satisfaction and vice versa. Orthogonal Frequency-Division Multiplexing (OFDM) is a digital multicarrier modulation scheme in which a signal is partitioned into several subchannels at different frequencies. OFDM-based systems are able to deliver a high data rate, achieve high spectral efficiency, operate in the hostile multipath radio environment, and reduce power consumption. Given these advantages, OFDM is becoming a fundamental technology in wireless communications and is now used in WiFi and WiMAX standards. For OFDM-based systems, we propose an efficient policy for controlling the admission of calls. The strategy will make an intelligent decision to either accept or deny a call request. It has been shown that the scheme is able to substantially enhance network throughput and increase service reliability.

15.2 Fast Recovery from Component Failures

A network-wide IP re-convergence is a time-consuming process, and a link or node failure is typically followed by a period of routing instability. Such a re-convergence assumes full distribution of the new link state to all routers in the network domain. When the new state information is distributed, each router individually calculates new, valid routing tables. During this period, packets may be dropped due to invalid routes. This phenomenon has been studied in both intradomain [3] and interdomain contexts [19], and it has an adverse effect on real-time applications [4]. Events leading to a re-convergence have been shown to occur frequently [32].

Much effort has been devoted to optimizing the different steps of the convergence of IP routing, that is, detection, dissemination of information, and shortest path calculation, but the convergence time is still too great for applications with real-time demands [10]. A key problem is that because most network failures are short-lived [22], a too-rapid triggering of the re-convergence process can cause route flapping and increased network instability [3].

The convergence process is slow, because it is *reactive* and *global.* It reacts to a failure after it has happened, and it involves all the routers in the domain. In the Resilient Networks project, we have developed methods for *proactive* and *local* protection mechanisms that allow recovery in the range of milliseconds. Our main contribution in this area is the use of a small set of logical topologies for protection. These topologies are constructed, so that a subset of the network is *isolated* (not used to carry traffic) in each topology.

Inspired by earlier work in interconnection networks [30], we developed a method for fast recovery in packet networks called Resilient Routing Layers (RRL) [11, 17]. Later, we refined this method and tailored it for use in IP networks, naming it MRC [16]. We describe its most important parts below. MRC has proven to be a flexible recovery scheme with a rich potential for optimizations in different respects. We have explored a number of improvements: protection against more than one concurrent failure [12], congestion-avoidance when traffic is sent on alternate paths [15], and a relaxation of the topology structure that provides improved performance in several ways [9]. We have also implemented MRC in a testbed of Linux boxes running the Quagga routing suite [2] and investigated some of the theoretical aspects of the creation of backup topologies [1].

An overview of MRC

MRC is based on using a small set of backup routing configurations, whereby each of them is resistant to failures of certain nodes and links. Given the original network topology, a *configuration* is defined as a set of associated link weights. In a configuration that is resistant to the failure of a particular node n, link weights are assigned so that traffic routed according to this configuration is never routed through node n. The failure of node n then only affects traffic that is sent from or destined for n. Similarly, in a configuration that is resistant to a failure of a link l, traffic routed in this configuration is never routed over this link; hence, no traffic routed in this configuration is lost if l fails. In MRC, node n and link l are called *isolated* in a configuration when, as described above, no traffic routed according to this configuration is routed through *n* or *l*.

Our MRC approach is threefold. First, we create a set of backup configurations, so that each network component is isolated in one configuration. We have developed several algorithms capable of creating limited sets of configurations, so that all links and nodes are isolated in arbitrary bi-connected topologies. Second, for each configuration, a standard routing algorithm, such as OSPF is used to calculate configuration-specific, shortest path trees and create forwarding tables in each router, based on the configurations. The use of a standard routing algorithm guarantees loop-free forwarding within one configuration. Finally, we have designed a forwarding process that takes advantage of the backup configurations to provide fast recovery from a component failure.

Fig. 15.1a illustrates a configuration whereby node 5 is isolated. In this configuration, the weight of the stapled links is set so high that only traffic sourced by or destined for node 5 will be routed over these links, which we denote *restricted* links.

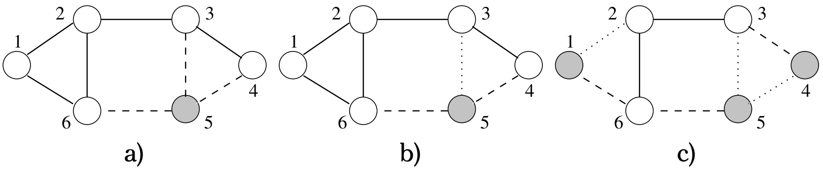


Figure 15.1 a) Node 5 is isolated (shaded color) by setting a high weight on all its connected links (stapled). Only traffic to and from the isolated node will use these restricted links. b) The link from node 3 to node 5 is isolated by setting its weight to infinity, so it is never used for traffic forwarding (dotted). c) A configuration where nodes 1, 4 and 5, and the links 1-2, 3-5 and 4-5 are isolated.

Node failures can be handled through blocking the node from transiting traffic. This node-blocking will normally also protect the attached links. But a link failure in the last hop of a path obviously cannot be recovered by blocking the downstream node (ref. 'the last hop problem'). Hence, we must ensure that, in one of the backup configurations, there exists a valid path to the last hop node, without using the failed link. A link is isolated by setting the weight to infinity, so that any other path would be selected before one including that link. Fig. 15.1b shows the same configuration as before, except now link 3-5 has been isolated (dotted). No traffic is routed over the isolated link in this configuration; traffic to and from node 5 can use only the restricted links.

In Fig. 15.1c, we see how several nodes and links can be isolated in the same configuration. In this type of backup configuration, packets will never be routed over the isolated (dotted) links, and only in the first or the last hop will they be routed over the restricted (dashed) links.

Several important properties of a backup configuration are worth emphasizing. First, all non-isolated nodes are internally connected by a subgraph that does not contain any isolated or restricted links. We denote this subgraph the *backbone* of the configuration. In the backup configuration shown in Fig. 15.1c, nodes 6, 2, and 3 with their connecting links constitute this backbone. Second, all links attached to an isolated node are either isolated or restricted, but an isolated node is always directly connected to the backbone with at least one restricted link. These are important properties of all backup configurations.

Using a standard shortest-path calculation, each router creates a set of configuration-specific forwarding tables. For simplicity, we say that a packet is forwarded according to a configuration, meaning that it is forwarded using the forwarding table calculated based on that configuration.

When a router detects that a neighbour can no longer be reached through one of its interfaces, the router does not immediately inform the rest of the network about the connectivity failure. Instead, packets that would normally be forwarded over the failed interface are marked as belonging to a backup configuration and forwarded on an alternative interface towards their destination. The packets must be marked with a configuration identifier, so the routers along the path know which configuration to use. Packet marking is most easily done by using the DiffServ Code Point (DSCP) field in the IP header. If this is not possible, other packet marking strategies, such as IPv6 extension headers or using a private address space and tunnelling (as proposed in [5]) can be imagined.

It is important to stress that MRC does not affect the failure-free original routing, that is, when there is no failure, all packets are forwarded according to the original configuration, in which all link weights are normal. Upon detection of a failure, only traffic reaching the failure point will switch configuration. All other traffic is forwarded according to the original configuration as normal.

Concluding remarks

Fast recovery mechanisms have received much attention from both industry and academia over the last few years. The Resilient Networks project has a solid track record in this field and, in a short time, has established itself as one of the leading contributors in this area. In addition to the MRC method discussed here, which is based on building multiple logical topologies in the network, we have also devised methods for single [8] and double [14] fault tolerance based on the Redundant Trees concept [23].

In the future, we will continue to follow up on the work in this area and contribute to new solutions in cooperation with our international collaborators.

15.3 Routing Homeostasis

In this section, we present our recently initiated work towards a more robust intradomain routing method [18]. The goal of this work is to explore how robustness to component failures and unexpected changes in the operating environment can be made a first-order design goal in the routing method. We take an evolutionary approach, in which we seek solutions that can be integrated into or built upon an existing routing infrastructure. Our work addresses the following question:

> Can we design a routing scheme that is primarily robust to both traffic load variations and topology variations, that is able to proactively avoid congestion when possible, that results in good routes in terms of delay, that is stable without introducing extra signalling load in the routing plane, and that is also incrementally deployable today?

In answer to the previous question, we present *Routing Homeostasis*, or simply "Homeostasis". We selected this name, because biological homeostasis is the mechanism by which organisms manage to maintain a robust and stable function, despite significant and unpredictable changes in their environments. In this section, we present the main design principles of Homeostasis and discuss preliminary results from a simulation implementation. The key objectives in Homeostasis are:

Robustness to load and topology variations. Instead of using a single best path, or the set of minimum equal-cost next-hops, we use multipath routing on *a limited set of loop-free, unequal-cost next-hops*. Multipath routing, together with an adaptive load balancing scheme, are the key ingredients of robustness in Homeostasis. Multipath can give rapid fault recovery upon topological changes, and it can absorb short-term overloads by using additional next-hops when needed.

Latency-based route selection. In Homeostasis, delay is a primary factor in selecting routes and in load balancing. We use propagation delay as our routing metric, and select up to **K** loop-free next-hops, representing the **K** shortest paths towards a destination. The reason we limit the number of next-hops to **K** is to limit the propagation delay of the resulting routes.

Proactive load responsiveness and congestion avoidance. Instead of *reacting* to increased queuing delays or packet losses, Homeostasis attempts to proactively *avoid* congestion. To do so, the load balancing module monitors the utilization of each selected next-hop that is currently used to route traffic. When that utilization exceeds a certain threshold, new flows are routed to the next available next-hop in terms of propagation delay. In this manner, higher delay routes are used only when needed; in light load conditions, traffic is routed through the single minimum delay path.

Stability in the control plane. Existing load-adaptive routing schemes suffer either from instability risks, or they introduce significant signalling load in the routing plane. To avoid these types of problems, Homeostasis does not rely on load-adaptive routing. Multipath routes are chosen based on propagation delays (a static metric that does not vary with load), while load balancing is performed on a local basis without any signalling between different routers. Of course, local load balancing is less effective than global load balancing in rerouting traffic upon congestion. We believe that the stability risks, or control overhead, of global load balancing schemes (such as load-responsive routing or online traffic engineering (TE)) are more important issues for network operators than are the advantages of such schemes in terms of capacity utilization.

Nonparametric operation. In general, it is difficult to make accurate predictions about the future traffic demands of a network. While historical measurements can offer an indication about the expected long-term traffic pattern, it is well-known that Internet traffic exhibits significant variation over a wide range of timescales. Recent developments, such as overlay networks, peer-to-peer applications and Intelligent Route Control make it even more difficult to obtain accurate traffic matrix (TM) estimates. Homeostasis is a “non-parametric” routing mechanism, in the sense that it does not require a TM estimate.

Deployment over existing routing protocols. Homeostasis can be implemented over existing intradomain routing protocols, such as OSPF, IS-IS, or Enhanced Interior Gateway Protocol (EIGRP), because it does not require additional communication between routers. The multipath routing tables can be constructed using the routing updates of any link-state or distance-vector protocol, as long as propagation delay becomes the metric of each link. The load balancing module can be implemented locally, as part of the forwarding engine’s functionality, and it does not require changes in the routing protocol.

Main mechanisms

The two main components in Homeostasis are the routing method that selects and installs the available routes for each destination and the load balancing method that assigns incoming traffic to one of the available routes.

Selecting routes We base our route selection on the propagation delays through each neighbour. This metric is selected, because we want to minimize the latency experienced by traffic, and because it is a stable property of the network that does not change with input load.

We say that a neighbour node is a *feasible next-hop* if the node is closer to the destination than the current node. A router installs up to **K** next-hops towards each destination in its forwarding table, corresponding to the **K** feasible next-hops giving the shortest distance in terms of propagation delay. The motivation for limiting the number of next-hops to **K** is twofold. First, we want to limit the amount of state information stored in the (expensive) memory used in forwarding tables. Second, we want to avoid using routes with very long delays. The strategy of communicating the minimum delay to the destination is similar to the approaches used by multipath distance vector protocols, such as DASM (Diffusing Algorithm for Shortest Multipath) [38] and MDVA (Multipath Distance Vector Algorithm) [31], but we differ by not installing all feasible next-hops in the forwarding table.

Dynamic load balancing. The selection of next-hops described above does not depend on the load in the network and changes only in response to topology changes. In our approach, all load adaptation takes place by adjusting the amount of traffic a router sends to each feasible next-hop. Here, we describe how this load balancing is performed. The load balancing method is designed to minimize delays under normal load, react to congestion based on local information only, and avoid reordering of packets belonging to the same flow.

The assignment of traffic is done based solely on a router's local view of the load situation in the network. Routers do not distribute any information about their loads to other routers in the network. Hence, our approach requires no additional signalling and can work directly with any existing routing protocol.

Importantly, the feasible next-hops towards a destination t are not treated equally by our load balancing mechanism. Instead, they are ranked according to their propagation delays. The basic idea is that we want to use the shortest path next-hop, as long as the utilization of this link stays below a certain threshold, which we call the *spilloverThreshold* θ. Only when the utilization of the shortest path next-hop exceeds θ, will traffic bound for t be sent on the second shortest path. When the secondary next-hop also exceeds this threshold, we will start using the third-shortest path, and so on, until the utilization of all available paths reaches θ.

The reasoning behind this strategy is that up to a particular utilization threshold, the queuing delay of a link is negligible. When utilization exceeds this threshold, queues might start to build up. We aim to avoid the unpredictability of queuing and congestion; thus, we prefer congestion-free, feasible next-hops, even if that means increased propagation delays.

To minimize packet reordering (with its adverse impact on TCP performance), we keep track of individual TCP flows and ensure that they are always forwarded over the same route. Load balancing is performed by assigning new (previously unseen) flows to the correct route selected by the load balancing algorithm.

Preliminary evaluations. To assess the possible benefits of our method, we have conducted flow-level simulations on inferred ISP topologies. We have focused on the ability to successfully route the offered input load, when the elements in the traffic matrix deviate significantly from their expected values. The results so far are promising, suggesting that Homeostasis is able to handle a wider range of input traffic than previous load balancing methods, while keeping end-to-end delays close to optimal. Other simulations show that Homeostasis is able to offer higher throughput than previous methods, while being robust to changes in the input load and topology [18].

Concluding remarks

Routing Homeostasis is a recently launched effort, focusing on how networks can be built with robustness as a primary objective, within the existing connectionless IP-routing architecture. Central in our approach are multipath routing and dynamic load balancing between the available routes. Our early efforts have been focused on defining the operating principles of such a solution and on illustrating the potential performance gains, with respect to robustness. Our current focus is on building a more theoretical framework around the stability and performance characteristics of our method. We will also perform more detailed performance analyses through packet-level simulations and prototype implementations in our lab network. In the future, we will explore ways to increase robustness through maximizing the multipath capability of each router.

15.4 Scalability of BGP Interdomain Routing

In this section, we describe some of our efforts in the area of scalability in interdomain routing using the BGP routing protocol. Our focus is on scalability in terms of the increasing rate of updates (churn) as the network grows. We seek to characterize the evolution of churn in the Internet, to understand the underlying reasons for this evolution, and ultimately to identify the most efficient solutions for limiting churn and hence improve scalability. Our work will contribute towards deciding whether BGP can continue to be the glue that binds the Internet or whether a completely new routing architecture will be needed to cope with a network the size of the future Internet.

To illustrate the growth and variability in churn, we plot the daily rate of BGP updates received from a RIPE routing monitor located in France Telecom's backbone network [27]. Figure 15.2 shows how the number of updates increased throughout 2005–2007. Using the Mann-Kendall test, we estimate an increase of 200 per cent in churn occurred over this three-year period. Projections of trends, such as this one from other networks, indicate that churn might grow to a rate that is unsustainable over the next decade [13]. The underlying reasons for this growth are not well-understood. A main goal of our work is to explain the evolution of churn and propose ways to improve scalability. Towards this goal, we have identified three dimensions that determine the level of churn experienced at different locations in the Internet. These are

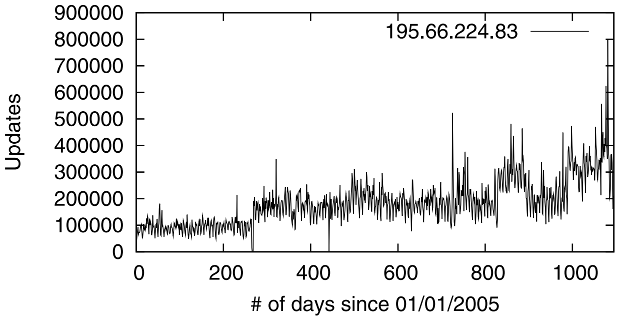


Figure 15.2 Growth in churn from a monitor in France Telecom's network.

1. **The structure and growth pattern of the Autonomous System (AS) level Internet topology.** Characterizing the Internet topology has been the subject of much research (and heated debate) over the last decade [6, 20, 21, 29, 39]. Today, we have quite a good understanding of several important topological properties and how they have evolved as the network grew. However, little previous work has focused on how these properties affect routing in general, and scalability, with respect to churn in particular. Examples of topological factors that influence churn are the hierarchical structure of the AS-level topology, the multihoming

degree (MHD) of networks at different locations in the hierarchy, and the use of settlement-free peering between networks.

2. **The various events and practices that trigger routing updates.** Routing updates are caused by changes in reachability for a prefix. The underlying reasons for such a change can be many (and complex). Some examples include broken BGP sessions, policy changes at an AS, intradomain routing changes that lead to changes in the preferred AS path, and topology changes. Anecdotal evidence also suggests that the use of BGP for inbound traffic engineering is responsible for much of the increase in churn throughout the last few years.
3. **BGP protocol options.** Although BGP is a rather simple path vector protocol at its core, many additional features have been proposed or standardized for increased flexibility, scalability, and performance over the years. Some examples include update rate-limiting with the MinRouteAdvertiseInterval (MRAI) timers [26], Route-Flap Dampening (RFD) [25], multipath extensions [7, 37], mechanisms for faster convergence [33, 36], or interdomain traffic engineering methods using communities, Multi-Exit Discriminators (MEDs), and selective subprefix advertisements. Such mechanisms increase the flexibility of interdomain routing (and are sometimes necessary for achieving policy-compliant routing), but they also increase the parameter space, and they make it difficult to predict the behaviour of the global routing system.

The effect of topology growth

We present sample results from our work exploring the first of the three dimensions mentioned above. In the work presented here, we look at several “what-if” scenarios for topology growth and investigate their impact on the number of update messages received at each AS after a well-defined event at the periphery of the network. A more detailed description of this work can be found in [35].

As a starting point for our investigations, we define a Baseline growth model that is similar to the growth seen in the Internet over the last decade [34]. Starting from this Baseline growth model, we investigate several single-dimensional deviations from this growth pattern and investigate the effects they have on the observed churn. We focus on events in which individual destination prefixes are withdrawn and then re-announced by the owner. This is the most basic routing event that can take place in the Internet and, at the same time, the most radical; these changes must be communicated all over the network. For different topology growth scenarios, we measure the number of routing updates received by nodes at different locations in the network. Next, we give some sample results from these investigations.

As an example, we look here at how the mix of different node types affects churn, by considering four different deviations from the Baseline model, with respect to the mix of nodes at the edges and in the core of the network. These deviations illustrate how economic factors might create a very different fauna of networks than those we see today. In this experiment, we change the fraction of stub, transit, and tier-1[1]

[1] Tier-1 nodes are defined as nodes without providers.

nodes, while keeping all other characteristics unchanged from the Baseline growth model.

NO-MIDDLE In the first deviation, we look at a network with only stub and tier-1 nodes. This illustrates a scenario in which the price for transit services from the globally present tier-1 nodes is so low that they have driven regional transit providers out of business.

RICH-MIDDLE In the second deviation, we focus on the opposite scenario, in which the ISP market is booming, and there is room for a plethora of smaller transit networks.

STATIC-MIDDLE In the third deviation, we look at a situation in which all network growth happens at the edges of the network. The number of transit providers is kept fixed, and the network grows only by adding stub nodes. This could be a plausible future scenario, if the ISP population becomes stable.

TRANSIT-CLIQUE In the fourth and final deviation, we let all transit nodes be part of the top-level clique, that is, all transit nodes are tier-1 nodes. This scenario may seem far-fetched, but it is important, because it shows what would happen if the transit-provider hierarchy collapses to a clique of 'equals' connected by peering links.

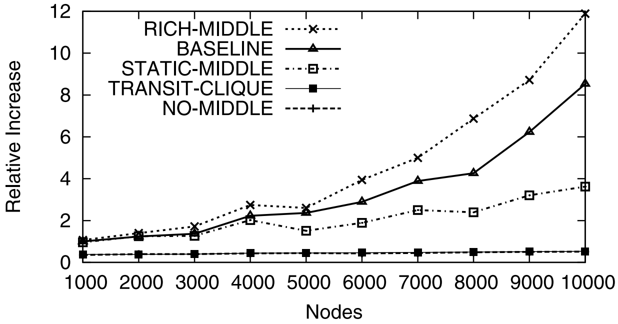


Figure 15.3 The effect of the AS population mix.

Figure 15.3 shows the increase in the number of updates seen at a tier-1 node after the withdrawal and re-announcement of a prefix from a stub node, as the network grows. The increase is normalized with the number of updates in the Baseline model at $n = 1000$.

A first observation from the graphs is that the node mix has a substantial influence on churn. In particular, the comparison of RICH-MIDDLE, Baseline, and STATIC-MIDDLE shows that the number of M nodes is crucial. We also observe that the number of T nodes in the network does not have any impact on the number of updates by itself. The only difference between the deviations NO-MIDDLE and TRANSIT-CLIQUE is in the number of T nodes, and we see that the number of updates is the same in these two scenarios. These observations are discussed in more detail in [35].

An important conclusion to be drawn from the above observations is that *the increased number of updates does not come primarily from an increased number of transit nodes, but from the hierarchical structure in which they are organized.* An Internet with several tiers of providers buying transit services from other providers gives a much higher update rate than a flatter topology in which most stub networks connect directly to tier-1 providers. Whether the Internet will move towards a more hierarchical or a flat topology in the future is difficult to predict. We do know, however, that the average path length, measured in AS-level hops, has remained roughly constant, at around four hops, during the last 10 years [34]. This implies that the Internet retains some hierarchical structure, and that the depth of that structure does not seem to vary with the size of the network.

Concluding remarks

Our efforts on BGP scalability so far have received good feedback, and we plan to pursue it in several directions. This work will be carried out in close collaboration with our colleagues at the Georgia Institute of Technology and the University of Massachusetts. In addition to the simulation-based work described here, we have started a measurement study to better understand the evolution of churn in the Internet over the last 6 years. Using BGP traces from publicly available data sets, we analyze the updates sent out by monitors in several tier-1 networks and seek to understand the driving forces in the evolution of churn. We also plan to develop analytical models for BGP churn. For this, we will use simple, regular topologies that capture the use of policies and the hierarchical nature of the Internet topology.

15.5 Packet Forwarding in Wireless Sensor Networks

We concentrate on an energy-efficient packet forwarding with QoS guarantee in WSNs. The key idea is to simultaneously consider the route planning and the transmission rate. We formulate the problem within a dynamic programming framework. Both theoretical analysis and simulation demonstrate the merit of the proposed strategy. It has been indicated that by deliberately trading off the range and rate of transmission, the end-to-end power can be significantly saved, and the end-to-end delay can be satisfied [43].

A WSN is modelled as an undirected graph. For an N-hop path $\{v_0, \dots, v_i, \dots, v_N\}$, v_0 and v_N represent the source node and destination node, respectively. All nodes are uniformly deployed in the region of interest. The data centre (i.e., the destination node in the network) has knowledge of the node density. Each node has its position information. The packet forwarding problem can be formulated in the framework of dynamic programming [40]. To accomplish packet forwarding, a packet should be routed from the source node to the destination node. This process is divided into a set of stages, which is in accordance with the hops along the routing path. At each stage, the state consists of two components: the position of the current node and the remaining transmission time. The second component is able to guarantee the QoS requirement with respect to end-to-end delay. In the routing path, each node needs to decide: how far away the next-hop node (or called *relay node*) should be and how much time should be allocated to forward the packet. Here, the decision does not indicate the real position of the relay node; instead, it describes the ideal position of the relay node. To find out the real position, we propose a procedure *relay-selecting algorithm* that works as follows: Given the ideal position, the nodes within the transmission range of the relay node are chosen. Among these selected nodes, the one nearest to the ideal position will be selected as the next-hop. In the dynamic programming formulation, a value function plays an important role. In our scenario, the value function $J_{\mu}(x)$ is defined as the average end-to-end energy consumption with respect to the initial state x and the policy μ. The objective is to find out the optimal policy. A policy μ^* is called optimal if $\forall x, J_{\mu^*}(x) \leq J_{\mu}(x)$ for every other policy. Let J^* denote the value function under the optimal policy. The optimal policy can be obtained by solving Bellman's equation [40]. Once the optimal value function J^* is available, the optimal forwarding policy is given by

$$\mu^* = \arg\min_{\mu} \left\{ g_{\mu}(x) + \int J^*(y) f_{\mu}(x,y)\, dy \right\} \tag{15.1}$$

where $f_{\mu}(\cdot,\cdot)$ represents the state transition probability density function, and $g_{\mu}(\cdot)$ represents the average one-hop energy consumption. It is observed that the optimal policy is composed of a series of optimal decisions made at each state x in order to minimize the right-side of (15.1). The key to solving the dynamic programming problem is to find the optimal value function. The value iteration approach can be employed to approach the optimal value function by continuous iteration in the value function space. As demonstrated in [40], the iteration process is able to converge into the optimal value function, given an arbitrary initial value.

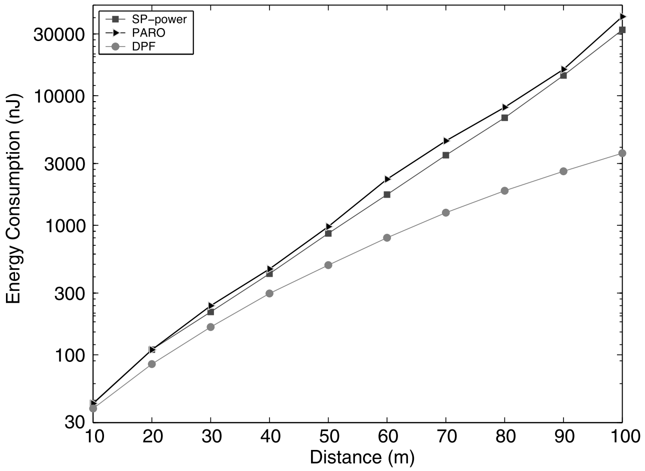


Figure 15.4 Energy consumption in terms of source-destination distance with difference schemes.

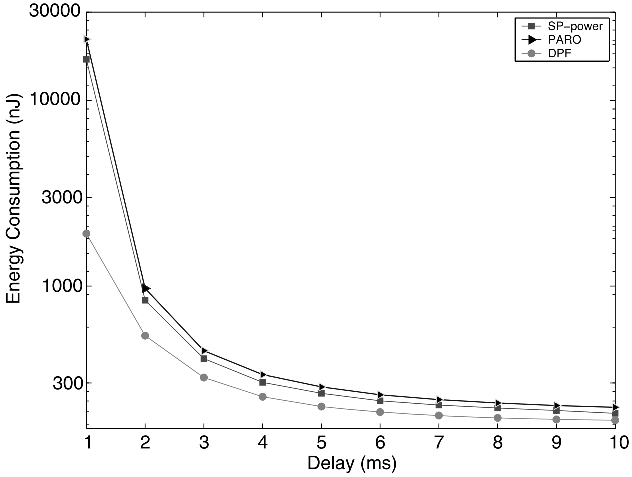


Figure 15.5 Energy consumption in terms of end-to-end delay requirement with different schemes.

A simulation result is presented to evaluate the proposed packet forwarding protocol. In the simulation, we consider a square field. Sensor nodes are placed uniformly at random in the region. We compare the proposed protocol with two previous energy-aware routing algorithms: SP-power [41] and PARO [42]. Fig. 15.4 and Fig. 15.5 show the energy consumption with respect to distance and delay, respectively. Our proposed scheme is denoted as DPF in the figures. It is clear that our proposed strategy is able to significantly reduce the energy consumption, especially when the end-to-end delay constraint becomes tight. This observation indicates the advantage in exploiting both temporal and spatial dimensions.

It has been widely accepted that the Multiple-Input Multiple-Output (MIMO) technique is able to remarkably improve the performance of a wireless system. With the recent advances of hardware design, MIMO can be integrated into a wireless sensor node such that WSNs performance can be greatly improved. From an energy efficiency point of view, it is beneficial to implement multiantennas in a sensor node. Each sensor node is equipped with isotropic transmission antennas and isotropic receiving antennas. The sensor nodes are sufficiently separated such that any mutual coupling among the antennas of different nodes is negligible. An efficient packet forwarding strategy is becoming critical in fully exploiting the benefit of MIMO in sensor networks. In addition, adaptive resource allocation is capable of significantly improving network performance under an application-oriented constraint. It is desirable to adaptively adjust the transmission power in each sensor node using a decentralized process. In the MIMO sensor networks, we study the packet forwarding protocol, taking into account channel coding, rate adaptation, and power assignment. This strategy considers the unique characteristics of MIMO, the physical layer, and also the routing layer, which is inherently a cross-layer approach. The objective is to guarantee the QoS requirement with limited power provisioning. The issue is again formulated within the framework of dynamic programming. In the MIMO sensor networks, the value function $J_\mu(x)$ is defined as the end-to-end successful transmission rate with respect to the initial state x and the policy μ. The optimal policy can be obtained by solving Bellman's equation [40]. Once the optimal value function J^* is available, the optimal forwarding policy is given by

$$\mu^* = \arg\min_{\mu} \left\{ g_\mu(x) + \int J^*(y) f_\mu(x,y)\,dy \right\} \tag{15.2}$$

where $f_\mu(\cdot,\cdot)$ represents the state transition probability density function. $g_\mu(\cdot)$ represents the average one-hop successful transmission rate. Fig. 15.6 shows the symbol error rate performance in terms of the normalized energy with different schemes. In this example, the end-to-end distance D is fixed as 70m. The results indicate that the proposed scheme outperforms the other two conventional algorithms, particularly when the total initial energy is small. This shows that power allocation is very important during path selection. Not surprisingly, at high normalized energy, the symbol error rate of all these schemes converges gradually as signal detections becomes accurate at high Signal-Noise-Ratio (SNR).

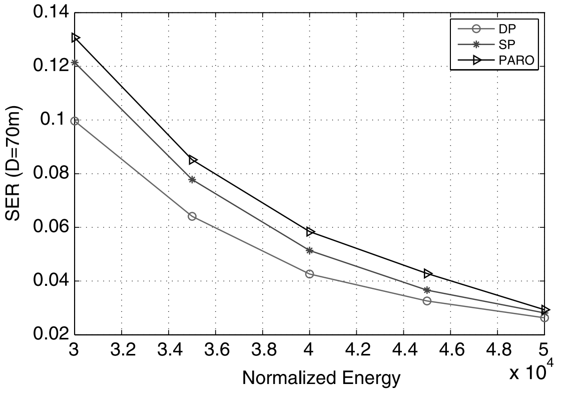


Figure 15.6 Symbol error rate (SER) in term of end-to-end energy constraint. The end-to-end distance is fixed as 70 *m*.

15.6 QoS in OFDM Wireless Networks

OFDM has been widely adopted in various wireless standards, for example, IEEE 802.11a/g for Wireless Local Area Networks (Wireless LANs), IEEE 802.16a/d/e for Wireless Metropolitan Area Networks (Wireless MANs), Digital Audio Broadcasting/Digital Video Broadcasting (DAB/DVB), and satellite radio [45]. OFDM is also the basis of Orthogonal Frequency Division Multiple Access (OFDMA), which is believed to be a highly promising technology for the upcoming fourth-generation (4G) wireless networks [44, 46]. In this section, we concentrate on the QoS, system modelling and tele-traffic analysis for OFDM wireless multiservice networks. Two different call admission control schemes, that is, a batch blocking scheme and a partial blocking scheme, are suggested upon subcarrier requests. We presented simulation results to validate the analytical model, and the two match each other very well. The QoS schemes, the analytical approach, and the results provide an efficient tool for evaluating next-generation broadband wireless networks [47].

In multiservice wireless networks, there are S classes of services. For each class-s (where $s \in \mathcal{S} \equiv \{1, 2, \cdots, S\}$), the batch arrival rate is denoted by λ_s. To provide service priority among various classes, we limit the maximum number of subcarriers employed by class-s to C_s ($s \in \mathcal{S}$). Upon the arrival of a call, if the number of free subcarriers is not less than the required number of subcarriers, the call is accepted. On the other hand, if the number of free subcarriers is smaller than the required number of subcarriers, the call is blocked directly; this mechanism is called a batch blocking scheme. Alternatively, the call can still be accepted with degraded QoS by employing the available subcarriers instead of the number of required subcarriers. This alternative mechanism is called a partial blocking scheme.

Blocking probability and bandwidth utilization are two important metrics in this system. In the batch blocking scheme, the class-s ($s \in \mathcal{S}$) call is blocked when either: 1) the summation of the requested batch size and the number of currently allocated subcarriers is greater than the maximum allowable number of subcarriers for class-s; or 2) the number of total free subcarriers is less than the requested subcarriers. The bandwidth utilization is the ratio between the average number of occupied channels and the total channels. In the partial blocking scheme, the call is blocked when either: 1) the number of currently used subcarriers of class-s is equal to the maximally allowed calls C_s; or 2) the number of currently utilized subcarriers of class-s is less than C_s, but all of the subcarriers are busy.

Illustrative numerical examples are presented to demonstrate the performance trade-off in the proposed call admission control algorithms. Without loss of generality, the system has two classes of services: narrow-band and wide-band calls. Fig. 15.7 shows the performance comparison between BBS and PBS in terms of narrow-band call traffic intensity. The narrow-band traffic intensity is defined as $\rho_n = \bar{x}_n \lambda_n / (C \mu_n)$. The solid curve and the dashed curve indicate the analytical results for BBS and PBS, respectively. The simulation results for BBS and PBS are, respectively, indicated by the circle symbol 'o' and the square symbol '□' for the purpose of validation. It can be seen that the simulation and the analysis agree very well. The blocking probabilities and bandwidth utilization increase as traffic intensity ρ_n increases. The average number of subcarriers used by narrow-band or wide-band calls decreases as ρ_n increases. We use BBS as the example with which to explain. With a higher ρ_n, more narrow-band and wide-band calls with a large bandwidth requirement will be blocked. Hence, the accepted narrow-band or wide-band calls exhibit the trend of lower subcarrier demand. Additionally, this will lead to a decreasing number of subcarriers used per accepted call. For either narrow-band or wide-band calls, the call blocking probabilities P_n and P_w in PBS are lower than the probabilities given in BBS. This can be explained as follows. In PBS, a narrow-band call or wide-band call is accepted, even if there are fewer free subcarriers than the number

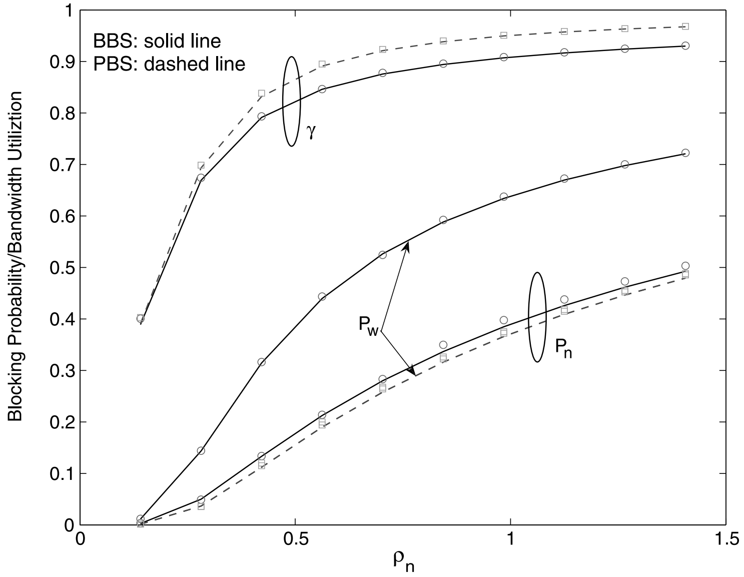


Figure 15.7 Performance comparison between Batch Blocking Scheme (BBS) and Partial Blocking Scheme (PBS). P_n and P_w represents the narrow-band and wide-band call blocking probability, respectively. γ represents the bandwidth utilization. The analytical results for batch blocking scheme and partial blocking scheme are represented by solid line and dashed line, respectively. The simulation results are represented by the circle symbol 'o' and square symbol '□', respectively.

of requested subcarriers. In comparison, in BBS, a narrow-band call or wide-band call is blocked when there are fewer free subcarriers than requested subcarriers. The enhancement in reducing the call blocking probability in PBS is clear with respect to P_w, while the reduction is insignificant for P_n. This is due to the much smaller batch size of narrow-band calls than that of wide-band calls. In addition, the comparison shows that the bandwidth utilization γ in PBS is higher than that in BBS. The reason is also due to the different policies used in BBS and PBS. The results suggest that BBS may be adopted for the case in which a shorter service time is preferable for accepted calls; PBS may be employed for a situation in which the call blocking probability is sensitive and can tolerate a longer service time.

15.7 Future Perspectives

Half-way through its funding period, the Resilient Networks project is well established in the area of resilient routing in IP and wireless networks, and has shown a high production of research papers in leading journals and conferences. In the next three years, we will continue along this current research direction, while expanding our efforts in some selected areas.

Our research studies on both BGP scalability and adaptive multipath routing were started in late 2007. We plan to continue and extend these two research threads for the rest of the project period. For the research on BGP scalability, we will continue our deep collaboration with our partners at the Georgia Institute of Technology. We are also in the process of starting a collaboration with colleagues at the University of Massachusetts, in this area. Working together with these academic partners, we will broaden the scope to include analytical modelling and experiments on real routers, in addition to more extensive and realistic simulations, while keeping the focus on the evolution of churn as the main area of study. In the area of multipath routing, we plan to develop our solution further, in order to maximize the ability to exploit the underlying path diversity, and to test and evaluate it more extensively under more realistic conditions. This will include implementing routing Homeostasis in our lab network. A longer-term goal is to study the implications of multipath routing on higher-layer protocols and applications.

It is widely believed that video content will dominate the traffic patterns of the future Internet. We believe that much of this video will be carried by ‘over-the-top’ applications that are not controlled by a network operator. Overlay networks will then play an important role. As we gradually phase out our research on proactive recovery mechanisms, we plan to shift our focus to methods that support communication between overlays and the underlying network layer with the goal of increased resilience.

Wireless resilience provisioning should take into account the unique characteristics and requirements in different protocol layers. In the physical layer, the main challenge lies in the unstable physical link. The emphasis should be put on channel modelling, signal processing, and cooperative communications. In the Medium Access Control (MAC) layer, resolving multiple access collision is a major concern to achieve maximum throughput. In the routing layer, multipath routing with resilient coding schemes may be an option to delivery packets as much as possible, with sufficient reliability and correctness. In the scheduling phase, transmitting packets in a fair and efficient manner is of the utmost importance. In addition, mobility and security management are basic components affecting all protocol layers. Thereafter, it will be advantageous to share information among different layers and optimize wireless resilience from a cross-layer perspective. Following this direction, we will propose a new energy efficient local metric for wireless sensor networks, considering both the maximum forwarding distance and the packet’s success probability. This will enable the forwarding node to choose the most energy efficient relay node in the routing protocol. Based on the new metric, we will present a cross-layer packet forwarding protocol by optimally selecting the relay nodes. It is envisioned that the new routing protocol consumes much less energy and generates significantly decreased signal overhead. The reduced energy consumption is able to significantly prolong the network’s lifetime and consequently the robustness against malfunction. As an initial effort, we reported a cross-layer optimized routing for sensor networks using dynamic programming [48]. In Cognitive Radio (CR) networks, we will propose a variety of joint designs, for example, joint scheduling/MAC/routing/topology control/power control strategies. We will emphasize the optimal and systematical perspectives. We will rely heavily on queuing theory, optimization, and game theory to increase system resilience and QoS in cognitive mesh networks and sensor networks with respect to energy, resource, mobility, spectrum, and data management.

The long-term goal of the Resilient Networks project is a network that offers a more resilient end-to-end service to the end user. This challenge requires carefully designed architectures and mechanisms on a range of different levels in end-systems, access networks and in the core network infrastructure. The complexity involved in this makes it important to build a research group with the a broad expertize in several areas of networking. In this respect, Simula's organization where each research project keeps a 10-year perspective on their work has been crucial in allowing the Resilient Networks project to reach its current status.

References

[1] T. Čičić. On basic properties of fault-tolerant multi-topology routing. *Elsevier Computer Networks*, 2008.

[2] O. K. Apeland and T. Čičić. Architecture and performance of a practical ip fast reroute implementation. *IEEE Global Communications Conference (GLOBECOM 2008)*, 2008.

[3] A. Basu and J. G. Riecke. Stability issues in OSPF routing. *Proceedings of SIGCOMM*, pages 225–236, San Diego, California, USA, Aug. 2001.

[4] C. Boutremans, G. Iannaccone, and C. Diot. Impact of link failures on VoIP performance. *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 63–71, 2002.

[5] S. Bryant, M. Shand, and S. Previdi. IP fast reroute using not-via addresses. Internet Draft (work in progress), Oct. 2006.

[6] H. Chang, S. Jamin, and W. Willinger. Internet Connectivity at the AS-level: An Optimization-Driven Modeling Approach. *Proceedings of ACM SIGCOMM Workshop on MoMeTools*, 2003.

[7] D. Walton, A. Retana, E. Chen, and J. Scudder. Advertisement of multiple paths in bgp. Internet Draft, expires January 2009.

[8] T. Čičić, A. F. Hansen, and O. K. Apeland. Redundant trees for fast ip recovery. *Broadnets 2007*. IEEE, 2007.

[9] T. Čičić, A. F. Hansen, A. Kvalbein, M. Hartmann, R. Martin, and M. Menth. Relaxed multiple routing configurations for ip fast reroute. *NOMS*, 2008.

[10] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure. Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review*, 35(2):35–44, July 2005.

[11] A. F. Hansen, T. Čičić, S. Gjessing, A. Kvalbein, and O. Lysne. Resilient routing layers for recovery in packet networks. *Proceedings of International Conference on Dependable Systems and Networks (DSN)*, June 2005.

[12] A. F. Hansen, O. Lysne, T. Čičić, and S. Gjessing. Fast proactive recovery from concurrent failures. *Proceedings 42nd IEEE International Conference on Communications (ICC 2007)*, Glasgow, UK, June 2007.

[13] G. Huston and G. Armitage. Projecting future IPv4 router requirements from trends in dynamic BGP behaviour. *Proceedings ATNAC*, Australia, dec 2006.

[14] S. Kini, S. Ramasubramanian, A. Kvalbein, and A. F. Hansen. Fast recovery from dual link failures in IP networks. *in submission*, 2009.

[15] A. Kvalbein, T. Čičić, and S. Gjessing. Post-failure routing performance with multiple routing configurations. *Proceedings INFOCOM*, May 2007.

[16] A. Kvalbein, A. F. Hansen, T. Čičić, S. Gjessing, and O. Lysne. Fast IP network recovery using multiple routing configurations. *Proceedings INFOCOM*, Apr. 2006.

[17] A. Kvalbein, A. F. Hansen, T. Čičić, S. Gjessing, and O. Lysne. Fast recovery from link failures using resilient routing layers. *Proceedings 10th IEEE Symposium on Computers and Communications (ISCC)*, June 2005.

[18] A. Kvalbein, C. Dovrolis, and C. Muthu. Routing homeostasis: towards robust, multipath, load-responsive routing. *in submission*, 2009.

[19] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. *SIGCOMM*, pages 175–187, 2000.

[20] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)*, 2007.

[21] L. Li, D. Alderson, R. Tanaka, J. C. Doyle, and W. Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications (extended version), 2005.

[22] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot. Characterization of failures in an IP backbone network. *Proceedings INFOCOM*, Mar. 2004.

[23] M. Medard, S. G. Finn, and R. A. Barry. Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. *IEEE/ACM Transactions on Networking*, 7(5):641–652, Oct. 1999.

[24] D. Meyer, L. Zhang, and K. Fall. Report from the IAB workshop on routing and addressing. http://tools.ietf.org/id/draft-iab-raws-report-02.txt, apr 2007.

[25] C. Villamizar, R. Chandra, and R. Govindan. Route flap dampening. RFC2439, nov 1998.

[26] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). RFC4271, Jan. 2006.

[27] RIPE's routing information service. http://www.ripe.net/ris/.

[28] University of oregon route views project. http://www.routeviews.org.

[29] G. Siganos, M. Faloutsos, and C. Faloutsos. The Evolution of the Internet: Topology and Routing. *University of California, Riverside technical report*, 2002.

[30] I. Theiss and O. Lysne. FRoots, a fault tolerant and topology agnostic routing technique. *IEEE Transactions on Parallel and Distributed Systems*, 17(10):1136–1150, Oct. 2006.

[31] S. Vutukury and J. Garcia-Luna-Aceves. MDVA: a distance-vector multipath routing protocol. *Proceedings INFOCOM*, 2001.

[32] D. Watson, F. Jahanian, and C. Labovitz. Experiences with monitoring OSPF on a regional service provider network. *ICDCS '03: Proceedings of the 23rd Interna-*

tional Conference on Distributed Computing Systems, pages 204–213, Washington, DC, USA, 2003. IEEE Computer Society.

[33] A. Bremler-barr, Y. Afek, and S. Schwarz. A. the bgp convergence problem. *Proceedings INFOCOM*, 2003.

[34] A. Dhamdhere and C. Dovrolis. Ten years in the evolution of the internet ecosystem. *IMC*, 2008.

[35] A. Elmokashfi, A. Kvalbein, and C. Dovrolis. On the scalability of bgp: the roles of topology growth and update rate-limiting. *CoNext 2008*, 2008.

[36] D. Pei, M. Azuma, D. Massey, and L. Zhang. BGP-RCN: Improving BGP convergence through root cause notification. *Computer Networks Journal*, volume 48, June 2005.

[37] W. Xu and J. Rexford. Miro: multi-path interdomain routing. *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 171–182, New York, NY, USA, 2006. ACM.

[38] W. T. Zaumen and J. Garcia-Luna-Aceves. Loop-free multipath routing using generalized diffusing computations. *Proceedings INFOCOM*, pages 1408–1417, San Francisco, CA, USA, mar 1998.

[39] S. Zhou and R. Mondragon. Accurately Modeling the Internet Topology. *Physical Review E, vol. 70*, 2004.

[40] D. Bertsekas. *Dynamic Programming and Optimal Control.* Belmont, MA: Athena Scientific, 1995.

[41] I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Trans. Parallel Dist. Sys.*, 12(11):22–33, 2001.

[42] J. Gomez, A. Campbell, M. Naghshineh, and C. Bisdikian. Paro: conserving transmission power in wireless ad hoc networks. *ICNP*, pages 24–34. IEEE, 2001.

[43] R. Yu, Y. Zhang, Z. Sun, and S. Mei. Energy and qos aware packet forwarding in wireless sensor networks. *IEEE ICC*, 2007.

[44] A. Jamalipour, T. Wada, and T. Yamazato. A tutorial on multiple access technologies for beyond 3g mobile networks. *IEEE Commun. Mag.*, 43:110–117, Feb. 2005.

[45] I. Koffman and V. Roman. Broadband wireless access solutions based on ofdm access in ieee 802.16. *IEEE Commun. Mag.*, pages 96–103, Apr. 2002.

[46] J.-C. Chen and W.-S. E. Chen. Call blocking probability and bandwidth utilization of ofdm subcarrier allocation in next-generation wireless networks. *IEEE Communications Letters*, 10(2):82–84, Feb. 2006.

[47] Y. Zhang, Y. Xiao, and H. H. Chen. Queueing analysis for ofdm subcarrier allocation in broadband wireless multiservice networks. *IEEE Trans. on Wireless Communications*, 7(10):3951–3961, Oct. 2008.

[48] L. Song, Y. Zhang, R. Yu, and W. Yao. Cross-layer optimized routing for wireless sensor networks using dynamic programming. *submitted to IEEE ICC 2009*, 2009.

[49] H. C. Styron. Csx tunnel fire: Baltimore, md. Technical Report USFA-TR-140, US Fire Administration, 2001.

16

FROM GILGAMESH TO STAR WARS

An interview with Carsten Griwodz by Dana Mackenzie

Carsten Griwodz joined Simula in 2005 and became the head of the Networks and Distributed Systems department in January 2008. As one of the leaders of the RELAY group, which focuses on resource utilization in distributed systems, he specializes in what he calls the "technical aspects" of video on demand. Nevertheless, his work also has what could be called a non-technical or recreational side. Several of his projects relate to online games, and he was the main driver behind Simula's involvement with the World Opera project, which is described below. In some ways the entertainment world is a better test bed for new technologies than more "serious" business or scientific applications. "It seems safer to experiment with games than, for example, with remote operating theaters, performing operations on humans," he says. "The scenarios are similar, but it's just so much more risky in that context." Nevertheless, it is easy to imagine that the same tools that keep computer games running smoothly today could also one day be used in telemedicine.

Most of the interview below focuses on the projects that are going on within the RELAY group. Towards the end of the interview we briefly discussed the Networks and Distributed Systems department as a whole, and Griwodz highlighted the ways in which the communication between subgroups in the department has improved over the last five years.

"I understand you are originally from Germany. What lured you to Norway?"

"After my *diplom* (the master's degree equivalent) in Germany, I found a job in an IBM research lab in Heidelberg, which was really exciting. It hooked me on research, until the day when they decided to close the lab. Then I wanted to find an equivalent job to what I had done in Heidelberg inside IBM, but for doing the same thing again

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_16,

I would need a PhD. So I went to Darmstadt to get a PhD. After that I wanted to continue in research, but somehow it was more interesting to go someplace other than the United States. I looked for a place that was not so warm. I don't like warm weather that much. So I looked in Canada and Norway and came to Norway at the end of 2000, as a postdoc and later an assistant professor at the University of Oslo."

"How did you find out about Simula?"

"Actually, I had just started at the University of Oslo when Simula was being established. The Komsys group, in which I was a postdoc, was to become one of the groups in Simula. But I read the contract, and it looked like an industry contract where you give up all your rights. I didn't want to sign that, so I stayed at the university along with some others who didn't like it as well. However, because of personal conflicts I eventually wanted to leave that group. Pål Halvorsen and I asked for permission to join the ND department formally, while remaining at the university. Eventually it changed and I came here full-time in 2006."

"Did you still have to sign the contract?"

"Yes, but by that point I had been working here for about a year as a university employee. I saw that the environment at Simula is so incredibly much better than the university. People have so much more freedom here. The terms of the contract, so far, have never really been used. In the end, you keep your freedom to do research."

"What were some of the good things about being at Simula?"

"Well, the university also has some good things, right? But there is no space. We had our offices, Pål and me. Our PhD students were sitting in an awful room at the other end of the building, and our masters students would only come from their PCs at home for meetings every once in a while. Here, at Simula, we were pointed to a lab and told we could use as much of this lab space as we wanted. Nowadays we have roughly 15 masters students. Ten of them sit here. The PhD students are all within shouting distance. Here we can build a group that cooperates. The size we have as the RELAY subgroup of ND would not have been sustainable at the university because we would have had to do all the supervision ourselves. Nobody helped each other because they never met each other.

"The fact that Simula is not part of the campus changes the feeling. Students feel like part of a research environment. Even the masters students feel more like researchers, whereas at the university everyone is in his own layer. When you are a student, then you talk only to your supervisor. Here we were able to build a group across hierarchical layers."

"So you have this project feeling, and everyone can talk with everyone else?"

"Yes. If I have problems, I can go to a masters student and ask. There is no trouble. It's such a different scenario, a totally different world. Having common lunches and informal talks instead of arranging a one-hour talk two weeks in advance is so different.

"There was another reason for us to move here. Our approach fit better with other research at Simula's ND department than that of the researchers who are now the ND department at the university. We have a bottom-up approach, developing algorithms and testing them, or using a theoretical model and then going through a simulation to prove it is better. The ND department at the university, the middleware group, has a different view. They look at concepts and then they go down to the implementation."

"When you say bottom-up versus top-down, what is the bottom and what is the top?"

"We build small pieces. We modify existing code and figure out if it contributes to an application. We may have an application idea, but we always look at the individual mechanisms, and the mechanisms are our goal. We don't try to build an environment and have the environment as our research topic."

"What are the advantages of your approach?"

"We would like to do quantitative improvements in our research. We would like an approach that allows us, either experimentally or by analyzing a simulation, to prove that we can improve on something that was there before. The top down approach is more conceptual and qualitative."

"Can you give examples of some of these quantitative improvements?"

"There is a nice little visual game by one of our PhD students showing how if you have a network system, a simple circular movement is going to look interrupted or hopping if the traffic is disrupted a little bit. If we apply our fixes, it gives a nice smooth circular movement. We have another experiment that is not so easily quantified. It's a test of transferring voice samples over Skype. If you have network trouble in the original protocol, the TCP version, the rhythm will get lost because some sound samples are delayed before they are played out. If you listen to music, it sounds skewed. The pitch will be correct, but the timing seems to be off. With our modification, it is still off, but a lot less frequently. You will experience lag (excessive latency experienced by users) about 80 per cent less often, for international communication, than without the modification."

"Does anybody actually play music over Skype?"

"No, but the phenomenon is more easily detectable by listeners if we play a sample. The original motivation was to improve computer games. But it works just as well if you have a remote display, if you have a thin client and you want to get your display from a remote server. Your mouse movement and your typing will be smoother."

"Have you had any interest from game companies in your work?"

"It never was formalized, but we did get traces from Funcom, a game company here in Oslo. They had at the time one large multiplayer game called *Anarchy Online.* Nowadays their largest game is about Conan the Barbarian (*Age of Conan: Hyperborian Adventures*). We got traces from *Anarchy* that gave us a lot of insights into how game traffic really looks. We had a lot of unexpected results. You'd assume that game traffic would be very sporadic. A player would act a little bit, you'd see long times of no traffic, and then the player would be active again. That turns out to be correct. We also expected that players would react to things that happened at the server, so you would get bursts from multiple clients, when all of them would try to interact with the game at the same time. That's not the case. The reason is that human reactions are on the same time scale as the network's latency. If someone in Germany and someone in Norway and someone in Italy play with a server in the U.S., the reaction speed would be the same but because of delays it would get smoothed out, so that the actions don't arrive at the server at the same time. That kind of ruined one of the things we wanted to investigate for them, which was how to fix that. So we had to think of other things.

"A more successful idea was a way to reduce packet loss. TCP works like this: You send packets from one server to another, and the other side sends an acknowledgement the whole way back. If you lose a packet, it needs to be fixed before the packets that come later can be delivered to the application. Now if you send lots of packets, the protocol will start a retransmission if the receiver has received the three following packets without receiving the one that got lost. But this doesn't work well for game traffic because it has too little data; we say it's too thin. So TCP uses a backup mechanism, a timeout to retransmit the last packet. Unfortunately, that halves the speed with which the sender is allowed to send. It's a safety mechanism within TCP that says, if I have to use the timeout backup, then the network must be really, really busy. Of course it's a wrong assumption in this case.

"We found that the company could buy a proxy server in a region with a dense population, like central Germany or the east coast of the U.S., and route all the communication with the clients in that region through this proxy. Then there will be only one thicker channel between the proxy and the main server, which will be better able to compete with the rest of the Internet traffic. Because the proxy is close to a lot of clients, local problems can be fixed a lot faster because the time distance between proxy and client is short. Between the proxy and the server, you have all of the client data in one channel, so the data stream is not so thin and the first mechanism of the three packets can be used.

"Although we never had any formal results from the discussion with Funcom, they introduced such proxies later. I think that we provided some additional motivation."

"Was this problem caused by TCP being used in a way it was never intended?"

"The funny thing is, it was intended for that, but nobody ever thought about applications that would care about such a fast reaction. TCP was supposed to allow you to log into a remote computer with a simple text shell and type. You do actually notice an improvement with our modifications, but in that context the problem is just not so bad that it matters very much. Especially in the last ten years, there has been so much focus on transferring lots of data over TCP that the improvements have really hurt the scenario with very few packets. So we are trying to raise the awareness that there is another problem in TCP that people should consider."

"One of your big interests is video on demand. How did you get involved with that?"

"I came to IBM as a protocols researcher, was placed in a group that built a video server, and never stopped doing that. I've done research on video on demand since 1993, and yet I have never owned a TV set!"

"Do any of your current projects relate to video on demand?"

"Yes. We are participating in a large collaboration that is partially funded by the Research Council of Norway as a Center for Research-Based Innovation. It is a project directed by the company Fast Search & Transfer, which has recently been bought by Microsoft. We are working on a prototype to figure out how we can integrate search for clips inside videos, in such a way that it works very efficiently and with high quality. For example, the user can search for all the scenes with a particular person and make his own little collage of video clips online, live and streaming from the server. For us it's not the search but the delivery part that's interesting. Search is for the other participants in the project. We try to move around the bits so that they come in time to the client to be played as video, getting the best possible quality to make it an interesting experience without interruptions. It looks very doable. We have not been working very long on this project; we just started two months ago."

"I would also like to ask you about the World Opera, which I see is part of the Verdione project. How did you get involved in that, and what are you doing?"

"There was a call by the Research Council of Norway in 2007 for large-scale grant proposals from groups of institutions, somewhere between a normal project and a research centre. These were called Stor-IKT projects, 'Stor' being Norwegian for 'big.' We brainstormed what we could do. Some of the suggestions were health services, sensor networks and so on.

"But there was a humanities professor, Niels Lund, attending the teleconference on the Tromsø site. He has this vision of building a world opera, where you would have at least two stages, with local singers and local orchestras and local audiences that would be recorded and displayed on the other end. Of course, because he was a musician, the audio was more important to him. However, the video delivery was extremely interesting to me, and somehow I just shouted louder than everybody else, so that became our topic for the application. We were one of the four Stor-IKT projects that were funded, and since then we have been trying to start everybody up.

"Simula's contribution to the project involves two main activities. One of them is to support the network, for the huge datastreams that need to be transferred during such a performance. Telenor is an active partner in this part of the project. Second, we have people here working on the video recording part. How do you extract a person from the stage and transfer this person in such a way that you can render him in 3D on the opposite side? And not only one person but all the people who are in the performance? The little difficulty is that the 3D displays that exist now are large; they make incredible noise that is not acceptable in opera; and we cannot afford them. So we are going to do something much cheaper on the rendering side, but that does not prevent us from developing the technology for all the rest.

"What we will be able to do, if nothing goes wrong, is to extract people from the stage and transfer all the data that is needed to render them from any angle. Unfortunately, with the currently affordable display technology, we will have to choose a very limited number of angles instead of showing a hologram. *Star Wars* isn't there yet! We are trying to get EU money together with a company that has been building holograms on a small scale, but we'll just have to see about that."

"When is the performance going to occur?"

"The idea is to perform an opera called *Gilgamesh*, which was actually written in 1974 for distributed stages. Of course, the author (Rudolf Brucci) was more thinking about television screens. We should be ready in 2011. Niels Lund also has his own opera that he is arranging, based on an old Danish book, but that will probably take a while longer."

"Where will the opera be performed?"

"Well, I'm not sure, but the opera companies involved in the project are Copenhagen, Stockholm, Malmo, and Milano. That would be one set. A second set would be the Metropolitan Opera in New York, Toronto, and San Francisco. It would be cooler to combine them into one event, but there is this little problem of time zones, so it wouldn't be practical."

"Are there any other projects in your group that you find particularly interesting?"

"We have an activity we call AMPS (Asymmetric Multiprocessor Scheduling), in which four PhD students are working on systems that will make it easier to develop parallel programs. That is how you achieve processing power nowadays, by parallelizing.

"There seem to be two philosophies of large-scale parallel computing. Scientific computing researchers have huge problems that they want to solve in the shortest time possible. They are so huge that it is worthwhile to use lots of time for the creation of a great program. Then they rent some huge machines for exclusive use for several days to run this program. The alternative is called cloud computing. Certain companies, like Amazon and Google, own lots of computers, which they would like to make available for 'normal' people. These people are most likely not very proficient at creating good parallel programs, so the people who give them access to their computing cloud need to help them with simple tools, while running many of the jobs in parallel. So the philosophies are 'excellent code, exclusive use of expensive resources' versus 'simplistic code, sharing abundant and fairly cheap resources.'

"We would like to do something in between, where you take parallel computing resources from the fantastic graphic cards that everybody has on his PC at home nowadays, take them to Amazon's clouds and make it possible to program something more complicated like video encoding or weather forecasting or games or search pre-processing. That's our goal. That will take a while."

"Can you narrow it down to some particular tasks that your students are working on?"

"An example is video coding. When you get video from a camera, you are getting 25 frames per second in Europe. To compress the file, you take one image and encode it, and take the next one and check how much difference there is with the previous one. You try to encode only the differences. That means your flow is both an encoding flow and one that goes back to the input to make the comparison. That gives you a nice graph of individual steps which have loops. It is really difficult to parallelize, especially with newer codecs with not just 2 pictures but 16 or 32 that you need to compare in all possible ways.

"If you have very low quality video, like video from a mobile phone, you can encode it on a normal computer. If you want to have high quality video like on a DVD, it may take a day to compute. But if you use the parallel resources of your computer, you might cut it down to twice the time that playback of the video takes.

"Now if you do large-scale encoding, you need that dozens or hundreds of times with the same performance. How do you do that? We need conceptual help, because it's too hard to program that kind of stuff. That's what we try to approach through graph theory. We structure the task as a graph, explain how the system looks as a graph, and try to map the task to the system in the best possible way."

"What kind of graph theory problem do you get?"

"First, you can think of modeling both the resource graph and the application as graphs with weights on the nodes and edges. In the process graph, the nodes indicate the amount of processing that needs to be performed, and the links are the amount of data we need to communicate between nodes. In the resource graph, the nodes are the processing capacities of the different processors, and the links are both network speed and memory speed.

"The cost, which we are trying to minimize, depends on the amount of communication between computers. This communication takes time and consumes bandwidth on the cable between them. On the other hand, the cost of the communication between two processors in the same machine is nearly zero. So we need to distribute the load that is represented by the process graph equally between machines, while having minimal costs of communication

"As a graph theory problem, we need to figure out where to make the split–which parts of the graph should be on different computers. Also inside the computer, we want to determine which parts of the graph to put onto different processors. In other words, it is a minimal-cut problem. Such a problem is easy to solve as long as you have only two computers, but it gets NP-complete if you have more. So we need heuristics, because otherwise the decision procedure is too slow.

"Finally, because our network changes over time, we have to figure out how to migrate after the initial distribution. The solution changes over time depending on the complexity of the input. In the Verdione application, maybe somebody switches off the light on the stage, and suddenly we get totally different complexity in the screen from the left side of the stage to the right. The load will shift a lot and we will need to redistribute the graphs."

"How do graphic cards enter into the problem?"

"When processes communicate even within the same computer, one processor needs to write to memory so that the other one can read it. In a normal computer that takes little time. But now that people want to use graphic cards, that step is expensive, just as expensive as communicating between computers."

"Do the cards then count as a new computer?"

"Absolutely. A laptop these days may typically have two cores, while a work station might have eight. A high end graphic card has 128. They are very stupid cores, but there are lots of them! You can do very few things with them, but those things that they can do, they can do very well."

"So it seems as if not all of your work is algorithmic–some of it is very conceptual."

"We try to look for real-world problems and then we get our method from them. We are definitely not method-driven. Our world is multimedia applications, and if we see a problem, we either try to figure it out or we give up, hopefully very early so that we don't waste time. We don't necessarily stick to the application that we started with, but we always use an application as a motivator."

"As the head of the ND department, what do you see the department doing in the next five years?"

"I see that we are getting more overlap in interest areas among the different projects inside the department. For example, the ICON group, which works on interconnection networks, is looking at how to apply their ideas of resource use in data centres. Now that's interesting, because it's pretty much what we want to do with our graph network. Maybe the granularity is different, but we will still be able to talk about approaches, algorithms, and good heuristics. So we have an interface there.

"In the Verdione project, RELAY is working with REPAIR, a group that has worked on making IP routing resilient to network problems on different layers, from a little network inside a house to the Internet at large. They are going to develop algorithms for Verdione, so that we can transfer our data. It's a very nice collaboration possibility.

"Also, there are two groups, the SimTel group and the Resilient Networks group, that both work on cognitive radio networks. Those are wireless devices that can look for a frequency that is currently unused and use it for communication until the one who owns the frequency comes in and wants it. Then they back off and try to find another one. It seems that this is really going to happen. One group looks at the problem in a very theoretical manner, the other wants to implement and test it. So there is a fantastic collaboration."

"It must be a challenge for you to direct such a department. Is it hard to steer the projects you're not even involved with?"

"I don't try. The projects in ND have historically been individual projects with their own leaders who are pretty strong. In the last evaluation we got the comment from evaluators that ND looked like two departments. I don't think we look like that any more, because we have these collaborations. I like to stay informed, and of course I’m responsible for finances, but otherwise the collaborations are developing naturally on their own, and that’s the right way of doing it.”

17

RELAY — ON THE PERFORMANCE AND RESOURCE UTILISATION OF TIME-DEPENDENT LARGE-SCALE DISTRIBUTED SYSTEMS

Carsten Griwodz and Pål Halvorsen

Carsten Griwodz · Pål Halvorsen
Simula Research Laboratory

Carsten Griwodz · Pål Halvorsen
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_17,

PROJECT OVERVIEW

RELAY

Distributed multimedia technologies permeate nearly all computerized products that are meant for end-users. When we use Skype to call a friend. When we watch YouTube videos that are transferred in real-time from Californian data centers. When we use team chat to plan strategy with our partners in a multiplayer game. And when we use our wireless network at home to listen to music from our home entertainment center, we enjoy also results of multimedia research, and the influence of multimedia research doesn't stop there.

In spite of or even because of the general adoption of distributed multimedia, research challenges are abundant. Skype will not yet be able to handle all of the world's telephone calls and you won't rely on it for calling emergency services. YouTube doesn't deliver satisfying video quality and you won't use it to enjoy an evening in front of your home cinema. Gaming systems don't scale yet, and you won't use this technology to perform remote surgery. Before we can rely on this technology, we need to improve our understanding of users' demands, and build the operating system and networking technology to fulfil them.

Scientific Challenges

The RELAY project performs research that will open new opportunities, bringing existing ideas closer to reality and finding entirely new applications. RELAY's speciality is to address performance issues of distributed multimedia systems. We investigate distribution mechanisms and network and operating system support that provide quantifiable improvements in resource utilization and the user's experiences of the service. We have given particular attention to video streaming and to interactive applications such as games, Internet telephony, and remote desktops. Better video streaming makes the distribution of videos more affordable and adapting the quality of videos to users' needs, and better means for interactivity improves the distances and number of participants that are still comfortable for users.

RELAY's research is driven by visions of a better user experience for various real-life applications, and research problems are taken from real problems experienced in industry or society. RELAY's approach to research often begins with a study of existing systems to identify limitations, bottlenecks, and problem areas in distributed systems that impact the quality experienced by end users. Once identified, the approach is to determine whether the limitation can be removed or relieved. An experimental approach is most frequently taken, and potential enhancements are tested in the context of real systems.

Obtained and Expected Results

RELAY partners with companies for new products and makes code available to everybody for download. One of RELAY's main achievements in the first phase of the project has been a set of transport protocol modifications that greatly reduces the delivery time of data in a lossy network for popular applications such as games and IP telephony. These protocol changes are currently being tested in Linux and have the potential to improve the quality of entertainment and communication applications for millions of users.

We have also continued our work on video streaming that has been integrated into projects such as the Center of Research and Innovation iAd, and started to explore the principles of using heterogeneous processing machines such as the Cell processor and graphics processing units as part of distributed multimedia systems.

RELAY — ON THE PERFORMANCE AND RESOURCE UTILISATION OF TIME-DEPENDENT LARGE-SCALE DISTRIBUTED SYSTEMS

17.1 Introduction

More powerful end-systems and faster access networks make large-scale distributed multimedia applications such as streaming, gaming, and Internet telephony increasingly popular. These services have a wide range of strict requirements with respect to timely data delivery and bandwidth. It is therefore difficult to provide proper system support in a shared distributed system of increasing heterogeneity such as the Internet. Challenges lie in system performance, scalability and utilisation, and the user's perceived service quality, which is frequently called the quality-of-experience (QoE).

Resource utilisation in time-dependent large-scale distributed systems (RELAY) is a project in the Networks and Distributed Systems department addressing issues of performance and resource utilisation in such distributed systems. The project investigates distribution mechanisms and network and operating system support that provide quantifiable better support for applications. While our interests include distributed system performance in general, we have been looking primarily at multimedia systems. These systems require that their varying requirements are enforced on a shared, best-effort Internet, i.e., a heterogeneous environment with fluctuating resource availability. Thus, important research questions for RELAY include the following:

- Where is the bottleneck that inhibits performance and limits end-user satisfaction?
- How do system and network performance influence application behavior and end-user perception?
- How can various components at different system layers be built, enhanced and integrated to improve resource utilisation and increase performance?
- What is the measurable gain of system and network improvements?

Thus, we investigate system-level approaches that provide better support for a class of distributed systems in spite of the lack of control over the infrastructure. RELAY helps ensure the maintenance of existing services and the deployment of new ones. We design, implement, enhance, and evaluate mechanisms, algorithms, and tools to improve resource utilisation, increase throughput, reduce latency, and support soft quality-of-service (QoS). We integrate and combine mechanisms to get more scalable, less resource demanding, high-performance systems for time-dependent large-scale distributed multimedia systems. We consider architectural, kernel, and protocol support for reduced resource consumption in servers and intermediate systems, as well as algorithms for the allocation of data and functions to servers and intermediate systems, and investigate combinations of performance-enhancing mechanisms such that they do not counteract each other.

In this chapter, we first describe the way researchers in the RELAY project have accelerated their activities at Simula due to the Simula way of performing and supporting research. Second, we describe the research performed by the RELAY group. We focus on the areas where we have the most results but also briefly mention other areas. In particular, our approach spawns subtopics that can have scientific impact beyond the application class where we found them. We use our research towards increasing the network performance of interactive applications as an example. In this kind of application, low latency is of preeminent importance for the user experience and we show the steps towards understanding the sources of latency, the development of improvements, and their evaluation in terms of QoE.

17.2 Simula as a Research Accelerator: RELAY's History

RELAY [26] was established as a Simula project when Carsten Griwodz and Pål Halvorsen, two faculty members of the Department of Informatics at the University of Oslo, were invited to join the ND group at Simula in 2005. The transfer to Simula then took place gradually, starting with 20 per cent of the positions for senior researchers in January 2005. In April 2006, all the activities were moved from IFI to Simula and, since January 2007, senior project members have had 100%-positions at Simula.

At the time of joining Simula, the project members had several years of experience in video-on-demand systems but few current externally funded activities within system support for multimedia and in video streaming in particular. Additionally, activities within gaming support were initiated. With respect to resources, a single project funded by the Research Council of Norway, MiSMoSS [18], had been granted, and there were two doctoral students and a few master students in the group. Thus, the resource situation was quite dire, with the doctoral students sharing a room in a distant corner room at the IFI building and with enough lab space for only up to two students available through the grace of a colleague.

The move to Simula resulted in a massive improvement in cooperation between all the group members, including both senior researchers and students. With ample office space within easy walking (or even shouting) distance, as well as opportunities for common lunches and the use of Simula's common facilities, cooperation within the group transformed from rigid and not very well-functioning weekly meetings into a cooperative spirit and informal but productive discussions. The ND department also controls a large lab, and we were told to "use as much of the lab as you want".

Thus, we have now filled the lab and been able to co-locate all our researchers and students.

The opportunity to move to Simula and the provided resources have accelerated all our activities, both in teaching and in research. The group has grown, as more master and doctoral students have joined and we are now participating in several external research projects. Next, we describe the RELAY research areas and goals, our way of performing research and organising our group, quantifiable achievements, and teaching and industry collaboration before looking at the research results in section 17.3.

Research Areas and Stated Goals

The senior members of RELAY have worked for many years in the area of system support for multimedia applications, and we continue previous work on multimedia distribution systems and operating system support for multimedia. Up to now, most similar research (including ours) has focused on continuous media-on-demand support such as in video-, movie-, news-, and audio-on-demand applications. We will continue this track because of renewed industrial interest and the new research questions raised for mobile systems, but we have extended our investigations to large-scale interactive applications. Our interests here include finding better and more efficient solutions with respect to performance and resource utilisation. Concerning the performance of individual computers, we cover optimised operating system kernel components and operating system support for heterogeneous multiprocessors. The latter also includes the use of modern programmable subsystems such as network processors, graphics processors, and physics engines. Distribution issues that we investigate include proxy server placement, the allocation of data and operations to the nodes in a distributed system, and latency reduction in interactive systems.

The goals of RELAY are therefore to design, implement, and evaluate algorithms, mechanisms, and tools to improve resource utilisation and performance, increase throughput, reduce/hide latency, and support QoS mechanisms that take QoE into account. We consider, in particular, the following:

- How the resource utilisation of individual nodes can be improved by using kernel extensions and programmable subsystems.
- How the resource utilisation of distributed systems can be improved through migration or replication of both functionality and data.
- How individual mechanisms at various layers of the system can be integrated and tuned.

We aim to publish the results in scientific papers at high-quality conferences and in journals in the various areas of multimedia systems. Software is developed in cooperation with others and published according to project specifications, under an open-source license when possible.

The RELAY Way

The RELAY group likes to conduct experimental research. This means that we perform benchmarks on systems to find potential bottlenecks and that we build, enhance, and integrate components and perform real tests using real systems. We enhance the target operating systems, middlewares or applications, and use realistic workloads to evaluate our improvements. Several such improvements are built and and evaluated in the course of a project. At the end of every project, we determine the combination of improvements that provides the best QoE. This differs from an approach that is frequently taken, which is to optimise components separately in the (often false) hope that the sum of optimisations will have the best overall result.

RELAY is active in several areas of distributed multimedia systems, but the branches of RELAY's research fall together to form one big picture in the research projects iAD [35] and Verdione [3]. In order to achieve good overall performance, it is therefore important to be able to integrate the different parts.

Teaching

The leave of absence granted by the University of Oslo has given senior researchers at Simula the opportunity of reducing their teaching load. While this appears rather delightful, it would *not* be the most appropriate way to go for RELAY seniors. Multimedia systems research requires prototype development and experimentation for every single publication. Thus, a large amount of work, both with respect to implementation and experimentation, is performed before obtaining publication-quality results. In the view of RELAY members, this makes it absolutely necessary to continue teaching relevant courses as was done before the move to Simula.

Visibility through teaching and a certain appeal for students who are attracted by system-level "hacking" has brought many master students to ND's lab, where they spend a huge amount of their time working on their thesis research. As a matter of fact, the non-hierarchical, chaotic, communicative and informal atmosphere of the lab is so much more attractive to most master students that they prefer this rather noisy place to the available master student reading room. Thus, the continuation of teaching and the working environment at Simula along with research topics whose applications are easily understandable for beginning master students have provided RELAY with manpower for our research and publications from master theses.

Commercialisation, Innovation, and Industry Contacts

In RELAY, we currently have several industry contacts who we hope will adopt our results. The most extensive industry cooperation that we have at this time is with Fast Search and Transfer (FAST), Schibsted, and other industrial partners through the iAD project. The project aims to develop new search and data delivery solutions. We expect to be able to perform new research in iAD and contribute with our expertise in performance and architecture, both in FAST's search and delivery systems and in Schibsted's media house applications. A particular activity with these partners, which will field-test on-demand search and retrieval on streaming media, has just been initiated. Additionally, our cooperation with the startup company Netview Technology provides a means of commercialising our own research results in Internet video delivery. Since our cooperation accounts for a major share of Netview's development activities, they give our results an excellent exposure. In particular, Telenor is currently strongly interested in a commercial collaboration through Netview. Industry is also very supportive of our work towards the inclusion of the results of the research into the Linux kernel described under "Thin Streams" in section 17.3.

17.3 Scenario: Interactive, Time-Dependent Applications

For several years, we had conducted research into the area of video on demand. We had addressed questions of efficiency and scalability for video distribution in the Internet by looking at operating systems support, protocols, implementation options, scheduling algorithms, and distribution architectures. Although the topic is clearly far from exhausted, we felt that we needed to add a different angle to our research into multimedia systems, one that would bring a new scalability challenge. The one aspect of multimedia systems that is frequently needed but that we hadn't considered in our research was interactivity, meaning the real-time interaction of users with each other through a distributed system. Many types of distributed interactive applications promised fruitful research. Video-conferencing is an interesting application that is quite closely related to video-on-demand, but first, it doesn't comprise a scalability challenge in any realistic setting, and second, we wanted a field that would not mean an incremental change from video-on-demand, but one with a different angle and the potential for later combinations to cover a large topic area.

We decided to look at highly interactive applications that are not bandwidth hungry. Examples of such applications include audio conferencing, online games, virtual shopping malls, financial systems, distributed interactive simulations, and distributed operation theaters. Each of these imposes demanding restrictions on network latency. To enable satisfactory interaction in audio conferencing, ITU-T has defined guidelines for the one-way transmission time. These guidelines indicate that users begin to get dissatisfied when the delay exceeds 150 to 200 ms, and that the maximum delay should not exceed 400 ms [20]. For gaming, it has been claimed that acceptable latency thresholds before users start getting annoyed are approximately 100 ms for first-person shooter (FPS) games, between 100 and 150 ms for a racing game, 500 ms for role-playing games (RPG), and 1000 ms for real-time strategy (RTS) games [32, 29].

To explore the demands of these applications through measurements, as is common practice in our research field, we analysed packet traces from several examples in this class of applications. First, we turned to massive multiplayer online games (MMOGs), which is currently the largest class of interactive multiuser applications, many distributed over the entire world. For example, in April 2008, World of Warcraft exceeded 10 million paying subscribers, playing in several different virtual worlds. The most scalable existing MMOG is EVE Online, which has admitted more than 48000 concurrent users into a single virtual world in January 2009. The number of gamers who play online has passed 44 per cent in 2006, and the current number of MMOG players is estimated at 17 million[1]. With most of the games subscription-based, MMOGs are a multi-million Euro market. Similarly, audio conferencing over IP has a steadily increasing number of users; for example, in Q4 2008, Skype had 405 million registered users[2].

Every occasional or regular user of these applications has experienced latency that is apparent at the application layer, often called *lag* by gamers, and most probably been annoyed by it. To research more than trivial remedies for this situation, it was first necessary to understand the networking demands. Identifying the problem through the study of networking traffic is typical for networking and multimedia researchers, and several investigations are documented in the literature [11, 25, 21, 22, 23, 24]. Early studies aimed at creating traffic models that could be combined with models for streaming media and download traffic into a more accurate picture of Internet traffic. In contrast, we aimed at improving the latency situation for this class of applications.

We have experimented with various interactive, time-dependent applications. Examples include the FPS game CounterStrike that researchers at Portland State University collected traces for (their own observations are reported in [14]) and anonymised network traces of the massive multiplayer RPG game Anarchy Online from the company that developed and operates the game itself, Funcom. A summary of the traces is given in table 17.1, which for later argument also lists some applications with different characteristics.

Traffic-Shaping Approach

We performed the first investigation under two assumptions:

- that a highly interactive application emits bursty traffic, where times of high activity mean a relevant increase in traffic, and
- that streams between clients and server are correlated when users interact.

We tried to verify these assumptions by analysing packet traces (`tcpdump` files) from a popular CounterStrike(TM) server[3]. CounterStrike is an FPS, which we considered a sensible choice, since FPSes are more fast-paced than the other interactive applications and are therefore bound to exchange information very frequently. CounterStrike uses UDP[4], with an average upstream bandwidth of 1500 bytes per second and 20 packets per second. Upstream packet sizes vary from approximately 50 to 300 bytes. More details can be found in table 17.1 and in [14]. We divided a 1.6 GB trace into individual client-server conversations, or *flows*. We took the ten largest flows and considered all combinations of these whose intersection contained five minutes of traffic or more.

[1] www.mmogchart.com/charts

[2] en.wikipedia.org/wiki/Skype

[3] Thanks to Dr. Wu-chang Feng and his colleagues at Portland State University for their data.

[4] User Datagram Protocol.

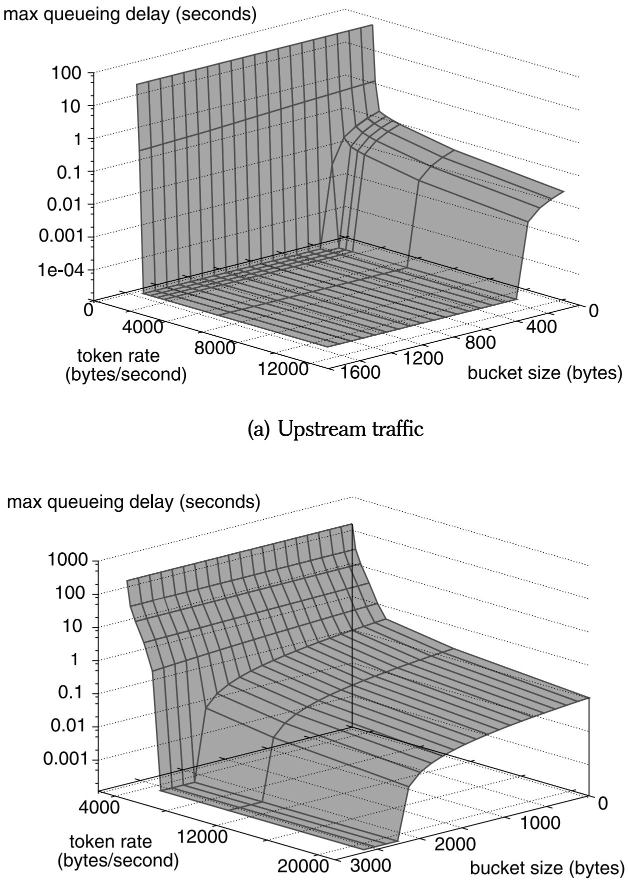
		Payload size (bytes)			Packet interarrival time (ms)						Avg. bandwidth	
Application	prot-								Percentiles		requirement	
(platform)	ocol	average	min	max	average	median	min	max	1%	99%	(pps)	(bps)
Anarchy Online (server side dump)	TCP	98	8	1333	632	449	7	17032	83	4195	1.582	2168
World of Warcraft (PC)	TCP	26	6	1228	314	133	0	14855	0	3785	3.185	2046
Age of Conan (PC)	TCP	80	5	1460	86	57	0	1375	24	386	11.628	12375
Counter Strike (PC)	UDP	36	25	1342	124	65	0	66354	34	575	8.064	19604
Halo 3 - high intensity (Xbox 360)	UDP	247	32	1264	36	33	0	1403	32	182	27.778	60223
Halo 3 - moderate intensity (Xbox 360)	UDP	270	32	280	67	66	32	716	64	69	14.925	35888
Gears of War (Xbox 360)	UDP	66	32	705	457	113	3	10155	14	8953	2.188	10264
Tony Hawk's Project 8 (Xbox 360)	UDP	90	32	576	308	163	0	4070	53	2332	3.247	5812
Test Drive Unlimited (Xbox 360)	UDP	80	34	104	40	33	0	298	0	158	25.000	22912
BZFlag	TCP	30	4	1448	24	0	0	540	0	151	41.667	31370
Casa (sensor/control system)	TCP	175	93	572	7287	307	305	29898	305	29898	0.137	269
Windows Remote Desktop	TCP	111	8	1417	318	159	1	12254	2	3892	3.145	4497
Skype (2 users)	UDP	111	11	316	30	24	0	20015	18	44	33.333	37906
Skype (2 users, TCP fallback)	TCP	236	14	1267	34	40	0	1671	4	80	29.412	69296
SSH text session	TCP	48	16	752	323	159	0	76610	32	3616	3.096	2825
YouTube stream	TCP	1446	112	1448	9	<1	<1	1335	<1	127	111.111	1278K
HTTP download	TCP	1447	64	1448	<1	<1	<1	186	<1	8	>1000	14M
FTP download	TCP	1447	40	1448	<1	<1	<1	339	<1	<1	>1000	82M

Table 17.1 Examples of packet traces for applications with *low-bandwidth, highly interactive* behaviour and for some **bandwidth-hungry** applications for comparison.

We wanted to find out whether we could merge or smooth flows in such a way that contemporary means of guaranteeing network resource availability could be used. The step needed for this would be to reduce burstiness.

Burstiness expresses how much data in a flow is sent in groups of packets that follow each other very quickly. It is said that these packets are sent *in bursts* and that they have a *low interarrival time* during a burst. This stands in contrast to non-bursty flows that consist of packets with equal interarrival times. Burstiness is typically expressed as some measure of packets' arrival-rate distribution, such as the coefficient of variation. We deal with flows which may have varying packet rates and packet sizes. In order to quantify the arrival distribution, we need to identify a relevant time interval Δt over which the arrival rate is measured. With bandwidth, which is typically measured in bits *per second*, one second is usually a reasonable choice for Δt. Highly interactive applications, however, require sub-second delays. Since these vary with the particular application, we use a simulated token bucket filter [28] to characterise burstiness independently of time scale. The token bucket filter models peak rate (how fast data can be processed for short periods, usually after saving some capacity), average rate (how fast data can be sent over long periods), and bucket depth (approximately the amount of capacity that can be saved without wasting any before using it for sending at peak rate).

We set the peak rate of the token bucket high relative to the arrival rate, resembling the line speed of a core Internet router. The token rate was varied from the average arrival rate (in bytes per second) found in the CounterStrike traces to 12 Mbps, simulating the amount of smoothing performed at a sender to avoid bursti-



(a) Upstream traffic

(b) Downstream traffic

Figure 17.1 Burstiness.

ness. The bucket depth was varied between 0 and 3000 bytes. For each setting, we measured the resulting worst-case queueing delay that would have occurred if the game trace data had been shaped by this token bucket. Figures 17.1(a) and 17.1(b) show the results for upstream traffic (client to server) and downstream traffic (server to client), respectively.

By plotting maximum queueing delay against the token bucket parameters, we obtain a visualisation of the flow's time-independent burstiness. For each scan line in the plot corresponding to a constant token rate, each drop-off in delay indicates a burst. Specifically, if a drop-off occurs at depth *d*, it indicates the presence of *d*-byte bursts. This is because a bucket of depth *d* allows bursts of up to *d* bytes to "pass through" at the (significantly higher) peak rate, so that they no longer contribute to the queue length.

The token rate of interest is the one at which, for some reasonable bucket depth, the corresponding queueing delay meets the application-specific delay bound. We notice that this happens for upstream traffic when the single largest packet of the trace is queued, meaning that a single packet can constitute the largest burst that is actually present in the data. Consequently, it is unacceptable to smooth the data flow from a client to a server to reduce burstiness, because the additional delay added by the smoothing would lead to a violation of the delay bound. For downstream traffic, the situation is slightly better, but the resulting observation is that the arrival of packets that are collectively as large as an Ethernet frame constitutes the largest burst in the data.

The inconvenient result of this investigation was that smoothing of individual streams is not feasible for an FPS (or any application with a similarly high need for interactivity, such as Voice over IP (VoIP) or remote controls), since a relevant smoothing does not occur unless packets are queued for more than the acceptable end-to-end delay. A different approach to performance improvement was necessary.

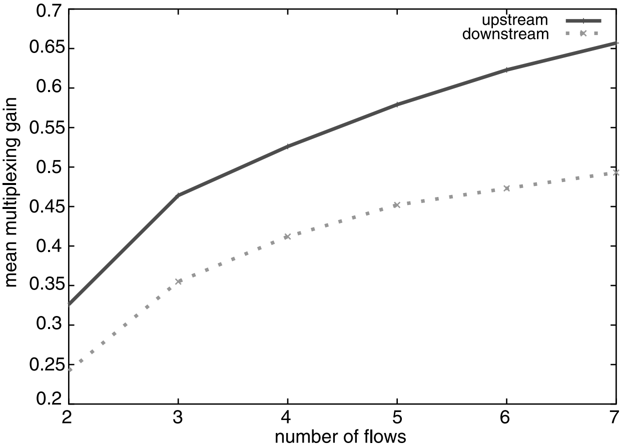


Figure 17.2 Multiplexing gain versus number of flows.

We quantify **correlation** between concurrent flows by running them through a simulated queue. We then look at the number of bytes per second that the network should transport reliably, the network's *service rate*. We call the smallest service rate of the queue for which a queueing delay bound Δ specified by the application is met the optimal service rate $\sigma(\phi)$. When we consider a set S of concurrent flows and compare their service rate $\sigma(S)$ with the sum of the individual flow's service rates $\sigma(i)$, we can measure the multiplexing gain mg achieved by combining the flows in S:

$$mg = 1 - \frac{\sum \sigma(i)}{\sigma(S)}.$$

A set of flows with mutually independent Poisson packet arrival distributions would yield a high multiplexing gain. Conversely, highly correlated flows would offer little multiplexing gain, because their peaks are likely to coincide. Figure 17.2 shows the improvement in multiplexing gain for the CounterStrike trace data as several flows are aggregated. This shows that significant multiplexing gain is possible, suggesting that peaks in individual flows are not strictly correlated in the aggregate. Multiplexing gain of downstream traffic is lower than that of upstream traffic, suggesting higher correlation between peaks in aggregated downstream flows. Knowing that the server sends out updates of the game state to clients cyclically, this is understandable.

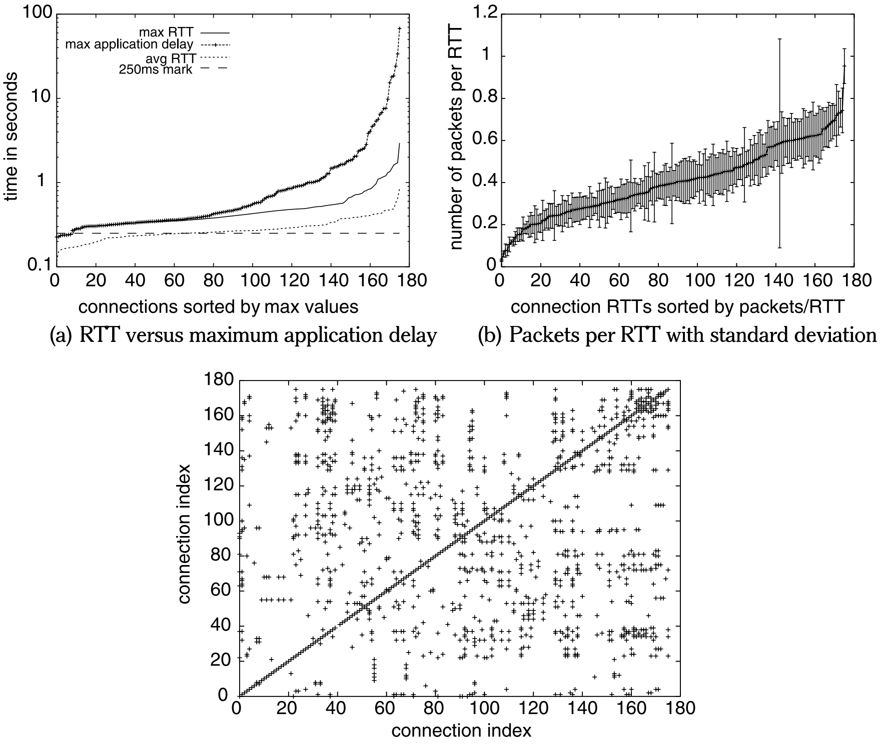
We expected that traffic in individual flows would show significant burstiness at time scales of one second or more, due to the interactive nature of FPS gameplay. This was not the case. It turns out that the number of packets that are exchanged between the server and an individual client within a duration that is equivalent to the acceptable end-to-end delay of the game is too low to show up as a burst. For the same reason, traffic shaping of individual flows is impossible.

Correlation between flows could probably be exploited, especially for games such as CounterStrike, where the server updates the game state cyclically. This works only in the downstream direction, however, since upstream traffic is hardly correlated; the apparent reason for this is that the difference in distances that different clients (in terms of network-layer latency) have to the server is on the same order of magnitude as the game's maximum worst-case latency. The differences are therefore overshadowing concurrent user action.

Thin Streams

The experiments described earlier in this section motivated us to take a closer look at the time-critical applications with very low network bandwidth consumption that seemed to be characteristic of many interactive multimedia applications. Because of their low bandwidth consumption and the small packets that they generate, we named them *thin streams*, in contrast to the frequently investigated greedy streams, which consume their share of the available bandwidth entirely and which we consider *thick*. These terms exist orthogonally to the established terms of *TCP mice* and *elephants*, which are associated with short-lived and long-lived TCP streams and which are not our concern here.

Funcom's network development team enabled us to take a closer look at a huge interactive application, Funcom's game Anarchy Online. Anarchy is a massive multiplayer game that thousands of players play at any given time. All Anarchy players meet and interact in a single virtual world (managed by a centralised cluster), in contrast to players of many other games, including World of Warcraft. In a one-hour packet trace that contains all packets from one of a few hundred game regions, we found approximately 175 distinct active TCP connections, and knowing that the servers are located in the United States, one can assume from the observed minimum latencies that there are players concurrently located in the US, Europe and Asia.



(a) RTT versus maximum application delay (b) Packets per RTT with standard deviation

(c) Matrix of connections probably sharing paths

Figure 17.3 Anarchy Online packet trace analysis.

Some of our findings are depicted in figure 17.3. From figure 17.3(a), we see that the average round-trip time (RTT) is somewhat above 250 ms, with variations up to one second, that is, these RTTs make the game playable [16]. Looking at the maximum application delay (time for receiving a successful acknowledgment), however, which may include several retransmissions as a result of packet loss, we find extreme delays of up 67 seconds. Obviously, we cannot distinguish between lost packets and lost acknowledgments in this server-sided trace, but we can see the potential for several second-long delays in delivering packets to the application on the client side. Furthermore, figure 17.3(b) shows that, on average, less than one packet is sent per RTT. Combined with the measured RTTs, we see that the number of packets per second is low, below two packets per second. Considering that each packet is very small (about 120 bytes on average), this demonstrates how thin the individual streams are. Finally, figure 17.3(c) shows that groups of these connections have a high probability of shared paths. It visualises the results of a Wilcoxon-rank test [13] of connection pairs, which checks whether their RTT values stem from the same RTT base value set. A dot in the figure shows that two connections share a path with high probability. We can see that several connections have a high probability of sharing several links in the path to the server, which indicates opportunities for bundling to conserve network resources[5].

With respect to losses, the original trace shows a loss probability of slightly less than one per cent but contains several instances of six retransmissions for the same packet. This implies that it is not the loss rate itself that is unacceptable, but the occasional huge delays when multiple retransmissions are needed. Moreover, we have not found any correlation between the losses of the various connections in the trace and could conclude that they are uncorrelated. This implies that losses are not due to server-sided bottlenecks. We use as a working assumption, however, that the thinness of the streams may actually hide loss correlation.

The lag that is experienced by players with lossy connections can be huge, and the thinness of the streams implies that retransmissions occur more or less only due to timeouts, since less than one packet is sent per RTT. Simultaneously, the TCP congestion window grows very slowly, even in TCP's slow start, since several writes will be merged into a single packet, and repeated losses can keep the congestion window size close to one. Thus, the main challenge for this kind of traffic is not the bandwidth, but the delay due to retransmissions that is typical when TCP is used for this kind of traffic. A frequent comment to these observations is that the developers made the wrong choice of protocol. Historical protocols, however, providing per-stream QoS guarantees have not become widely available. Moreover, protocols like UDP, which allow the sender to determine the transmission timing, are widely criticised for their lack of *congestion control mechanisms* and are often blocked by firewalls. Furthermore, applications like Skype that use UDP, in spite of said criticism, fall back to using TCP if UDP fails. Thus, TCP is and will be a frequently used protocol, and it is therefore important to note and deal with the most salient observations, which are as follows:

1. They hardly ever trigger fast retransmissions but mainly retransmit due to timeout and
2. TCP's congestion control does not apply, that is, the TCP stream does not back off.

We found many interactive applications that display this traffic pattern. Statistics for some are given in table 17.1. To formalise the term *thin stream* for flows that follow the pattern, we determined that they all share the following criteria: a) The packet interarrival time is too high to trigger fast retransmissions, and b) the packet sizes are usually far below the maximum segment size (MSS). In this case, lost packets are often recovered through timeout retransmissions, resulting in severely delayed data delivery [27]. All of these applications generate data in such a way that packets are small and/or have high interarrival time. All of them exhibit difficulties providing proper service when they are sent over connections with high RTTs using TCP,

[5] In [27], we did some initial experiments showing how this could be used for sending larger packets containing several application level units (destined to different users) to a proxy, but have not followed up on this after initial testing. However, after talking to Funcom about our initial results [27], they restarted their work on proxies and have proxies in use today.

because recovery mechanisms for packet loss that are designed for high-throughput applications are rarely triggered, since they never fill the allowed send window.

TCP Considerations

TCP Shortcomings

We observe that applications that generate thin TCP streams experience higher latency than other applications due to their failure to trigger fast retransmission. The reason is that fast retransmit, which enables retransmission of lost segments before a timeout occurs, depends on feedback from the receiver (ACKs). The mechanism requires three duplicate acknowledgments (dupACKs) to be triggered [9, 8]. Waiting until the third indication of loss was originally introduced to avoid spurious retransmissions when reordering takes place on the network. For thin-stream scenarios, where interarrival times between sent packets are relatively high, the consequence is that many, or all, retransmissions are caused by timeouts. This is because there are seldom enough packets in flight (packets on the wire) to generate the necessary feedback to trigger a fast retransmit. In addition, the retransmission timeout (RTO) increases exponentially when multiple losses are detected (exponential backoff), which results in a still faster latency growth. The mechanism is designed to ensure that an acceptable sending rate is found, and to prevent a stream from exceeding its fair share of bandwidth resources. Applications that provide no interactive service may do well under such conditions.

Applications that exhibit thin-stream properties, however, are often interactive and may suffer severely from the extra delay. They use their fair share only rarely and stay within the worst-case throughput of two packets per RTT most of the time. The large interarrival times make it impossible for thin streams to back off when packets are lost multiple times, resulting in a situation where the RTO value is very high without any actual benefit with regard to resource distribution on the network. The result for the thin stream is that the retransmission is delayed by seconds (or even minutes) if segments are lost several times. For this reason, we take a look at new mechanisms that are compatible with other TCP implementations but support interactive time-dependent thin-streams better.

TCP Enhancements

As a first step, we need to be able to distinguish between thick and thin streams. Where high-rate streams are only concerned with throughput, the perceived quality of most thin streams depends on the timely delivery of data. Working from the assumption that there is potential for large performance gains by introducing modifications that tune TCP for thin streams, we have tested the combination of several such mechanisms on typical thin-stream applications. We use the very conservative algorithm in figure 17.4 to decide when the stream is thin and thus when to apply the enhancements. The modifications are designed in such a way that they are transparent to the receiver, i.e., a server can run the modified TCP, and unmodified clients may still receive the benefits. Any receiver is able to receive the stream sent by the

```
if $\left( in\_flight \leq \frac{pttfr + 1}{1 - lossrate} \right)$
{
    apply modifications    /* thin */
} else {
    use normal sctp        /* thick */
}
```

Figure 17.4 Determining which mechanisms to use with thin stream detection.

modified sender, regardless of operating system and version. The modifications can also be used transparently for applications running on top of TCP. We implemented the following modifications in the Linux kernel (v.2.6.23.8)

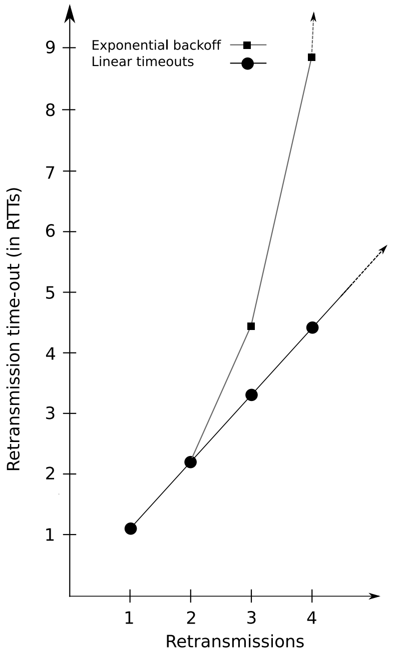


Figure 17.5 Difference between linear timeouts and exponential backoff.

Removal of exponential backoff: Since most thin streams consist of packets with high interarrival times, more or less all retransmissions are caused by timeouts. A timeout retransmission invokes exponential backoff, which doubles the time to wait

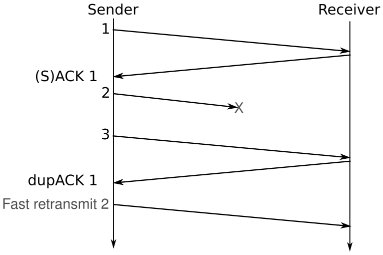
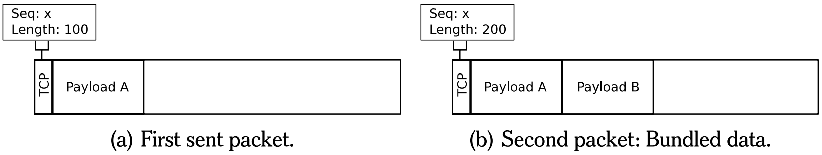


Figure 17.6 Fast retransmission upon first indication of loss.

for the next retransmission. If the number of packets in flight (i.e., unacknowledged packets) is less than the number required to trigger a fast retransmission, we remove the exponential factor for a behaviour as shown in figure 17.5. If more than four packets are on the wire, the possibility for triggering a fast retransmit increases, and therefore, exponential backoff is employed as usual. Since the streams that gain from this modification are very thin, the increase in bandwidth consumption due to the removal of exponential backoff in these cases is very small; that is, the stream still does not have to use its allowed send window.

Faster Fast Retransmit: Instead of having to wait several hundred milliseconds for a timeout retransmission and then suffer from the exponential backoff, it is much more desirable to trigger a fast retransmission. Normally, this requires the connection to wait for three duplicate acknowledgments (four acknowledgments of the same packets), which is not ideal for many thin stream scenarios. Due to the high interarrival times in our scenario, sending three packets often takes longer than the timeout. We have therefore reduced the number of required duplicate acknowledgments to one as shown in figure 17.6, provided that the number of packets in flight is less than four. Otherwise, the chance of receiving three dupACKs increases, and regular (three dupACKs) fast retransmit is employed.



(a) First sent packet. (b) Second packet: Bundled data.

Figure 17.7 Bundling unacknowledged data.

Redundant Data Bundling: As shown in table 17.1, many thin-stream applications send small packets. As long as the combined packet size is less than the maximum segment size, we copy (bundle) data from unacknowledged packets in the send buffer into the new packet. If a retransmission occurs, as many of the remaining unacknowledged packets as possible are bundled with the retransmission. This increases the probability that a lost payload will be delivered already with the next packet. Figure 17.7 shows an example of how a previously transmitted data segment is bundled with the next packet. Notice that the sequence number stays the same while the packet length is increased. If packet (a) is lost, the ACK from packet (b) will ACK both segments, making a retransmission unnecessary.

As mentioned, the first two modifications are only applied when there are fewer than four packets in flight. In this way, we avoid that streams that already use their fair share of the available resources (in accordance with TCP) can be stopped from consuming even more using our proposed mechanisms. Redundant data bundling (RDB) [33], on the other hand, is limited by the interarrival times and packet size of the stream. If the packets are large, which is typical for bulk data transfer, bundles cannot be made, resulting in normal TCP operation.

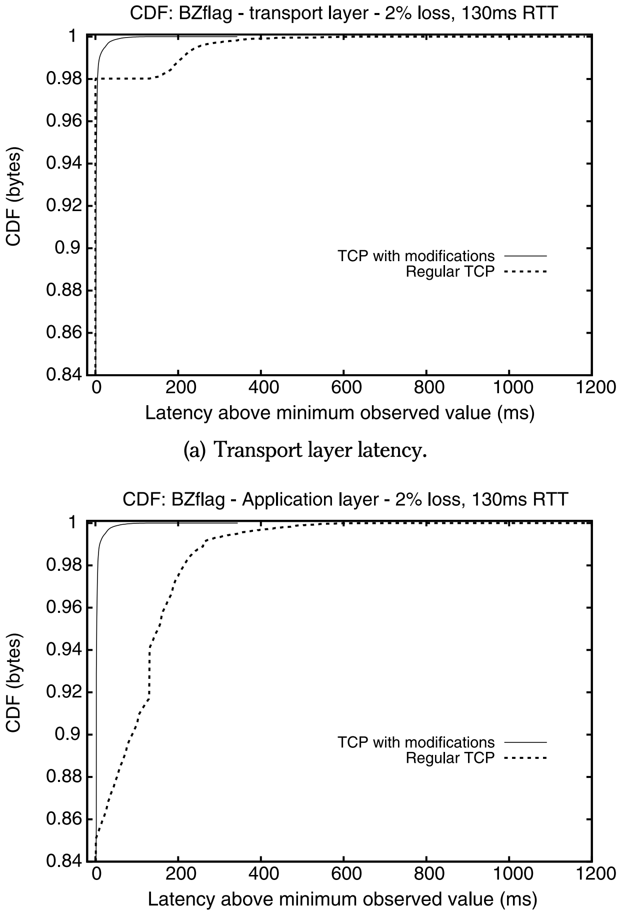
Experiments

The thin stream modifications are all implemented in the Linux kernel, and we have performed several experiments to measure the performance improvements, i.e., with respect to transport layer and application layer latency, and the quality of the user experience. To emulate a network, we used a machine imposing loss and delay on the link between the server and the clients. After performing several measurements from different Norwegian Internet service providers to machines in the United States and various European countries, we chose an average loss rate of two per cent and an RTT of 130 ms as representative emulated network values. In the next subsections, we present results from three different applications having the properties shown in table 17.1, that is, games, audio conferencing, and remote terminals using BzFlag, Skype, and SSH as representative examples, respectively.

Games: BzFlag

The TCP modifications are intended for interactive applications. To evaluate their effect, we benchmarked the position updates of a game. This was naturally not possible in a deployed commercial MMOG. We ran our experiments in BzFlag, an open-source FPS game where players challenge each other using tanks on a virtual battleground. As mentioned in table 17.1, BzFlag generates thin streams with an average packet interarrival times of 24 ms and an average payload size of 30 bytes. Thus, it is a game well suited to demonstrate the benefits of our thin stream modifications to TCP.

To collect the data needed to generate the results, we constructed a network consisting of five machines. Three acted as clients running computer-controlled opponents (bots), one ran the server application, while the fifth machine acted as a network emulator. The three clients ran 26 bots all together, which represents a typical high-paced BzFlag multiplayer scenario. To get a sufficiently large number of samples, we ran six one-hour tests (three with the modifications enabled and three without).



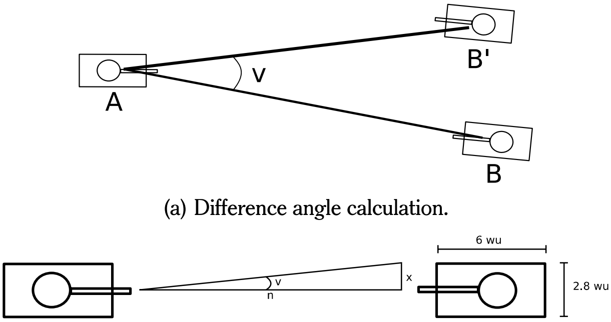
(a) Transport layer latency.

(b) Application layer latency.

Figure 17.8 Cumulative density functions (CDF) of BzFlag latency.

Latency results. From the packet traces, we measured how much later than the one-way delay (OWD) each data segment arrived at the receiver. Figure 17.8 shows cumulative density functions (CDFs) of the data delivery latency. The transport layer latency is shown in figure 17.8(a). Both connections received 98 per cent of the data in minimum time, meaning the only delay experienced was the OWD. This reflects a loss rate of two per cent. When the connection experiences loss, however, the modified system recovers the data significantly faster than the unmodified TCP.

The TCP in-order requirement states that received data segments must be stored in a buffer until a complete range of data are received before the data are delivered to the application. This means that successfully delivered packets still have to wait for any lost segments to be retransmitted and received before the application can use



(a) Difference angle calculation.

(b) Hit limit calculation

Figure 17.9 Calculations.

them. Figure 17.8(b) shows the application layer latency that takes this into account. We can see that much smaller latencies were experienced when the modifications were enabled. Close to 99 per cent of the data was available to the application at the best possible latency, compared to only 85 per cent when using ordinary TCP (a large increase, since only two per cent of the packets are delayed at the transport level). Unmodified TCP is not able to deliver 99 per cent in less than 261 ms. It is also worth mentioning that the modifications in this case ensure that the differences between application layer delivery time and transport layer delivery time are minimal.

Impact on user perception. The reduced application layer latency also affected the QoE. To see how the latency influenced the estimated player positions, we collected the actual and perceived position of the other players each time a chosen tank (reference tank) fired a shot. We then calculated the difference (angle) between the two positions as viewed from the reference tank. Figure 17.9(a) shows how the angle v between the two vectors representing them was found. Position A represents the reference tank, B represents the perceived position of an opponent at the time of the shot, while B' represents the actual position of the opponent. Figure 17.10 shows that the angle between The estimated and actual positions is smaller when the modifications were applied. On average, the angle between the estimated and actual positions was reduced by 0.7 degrees when the modifications were enabled (from 3.5 to 2.7 degrees). Provided that the player aimed at the centre of the opponent's tank (a "perfect shot") based on the estimated position, the angle between the estimated and the actual position might be so substantial that a would-be hit actually evaluates as a miss. Figure 17.9(b) shows how we calculate the deviation in world units (wu) caused by the deviation angle v. We can extrapolate the deviation in wu when the distance n to the target increases. Here, n is the distance between the player and the opponent, and x is the deviation from the actual position of the observed tank. In BzFlag, each tank is 2.8 wu wide and six wu long. A perfect shot would have a 100 per cent chance of hitting moving enemies when the distance to the target is less than 30 wu using the modifications. Using regular TCP, the distance to guarantee a hit would have to be reduced to 23 wu. In practice, this means that

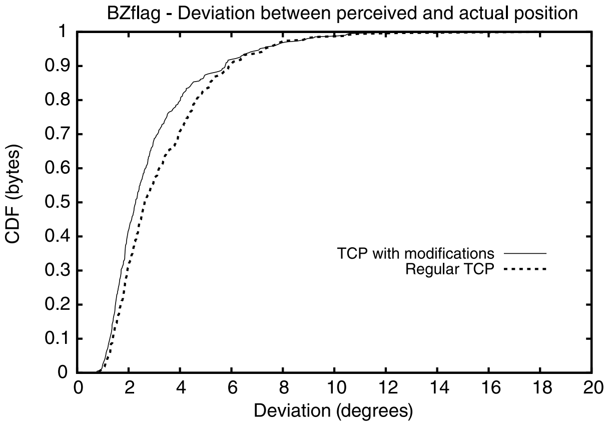


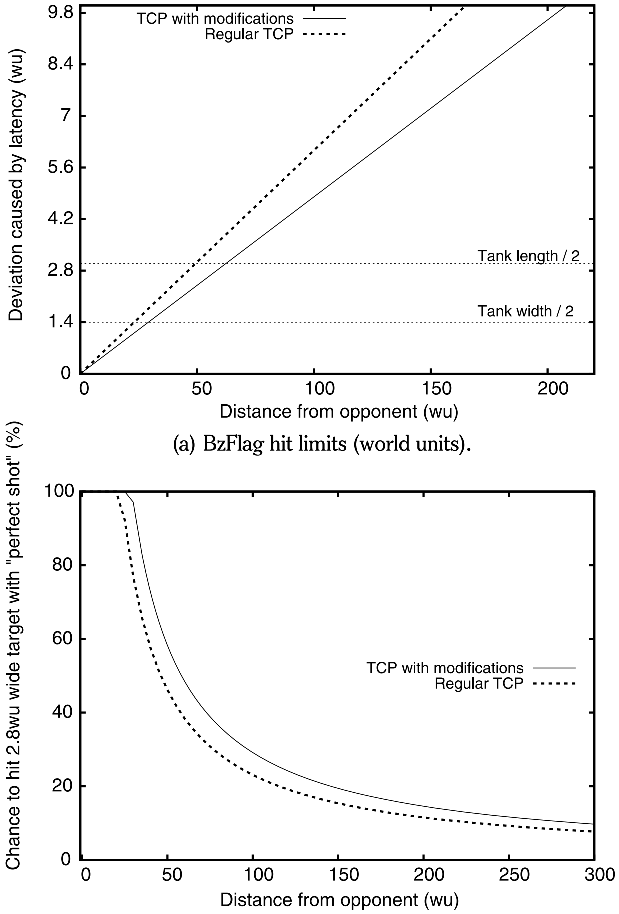
Figure 17.10 CDF difference.

the modifications increase the chances of hitting an opponent due to a smaller deviation between estimated and actual positions. The seven *wu* improvement is, for reference, equal to the width of 2.5 tanks in the game, which is large when trying to hit a target. We also investigated how the difference in angle affects the chance of the shot hitting when we varied the distance between the tanks. Figure 17.11(a) shows how large the radius of the hit box has to be to guarantee that the perfect shot is a hit given a specific distance to the opponent. The dotted lines at 1.4 *wu* and 3 *wu* represents the tank hit box when seen from the front and from the side. The graph shows how the effect of the modifications increases with the distance to the opponent being shot. Figure 17.11(b) shows the chance of hitting an opponent with a perfect shot at different distances using regular TCP and the modifications.

Audio Conferencing: Skype

Audio conferencing with real-time delivery of voice data across the network is an example of a class of applications that uses thin data streams and has a strict timeliness requirement due to its interactive nature. Nowadays, audio chat is typically included in virtual environments, and IP telephony is increasingly common. Due to efficient audio codecs and latency limited by QoE requirements, they are thin-stream applications.

Skype [6] is a well-known conferencing service with several million registered users who communicate in pairs or small groups on the Internet. We have analysed several Skype sessions and seen that this application shares the characteristics of IP telephony; the packets are small and the bandwidth low. Table 17.1 shows two examples of analysed dumps from Skype conferencing traffic. Using UDP, the packet payload size for our trace is very low, with a maximum payload of 316 bytes. The interarrival time between packets averages to 30 ms, which qualifies it as a thin-stream candidate. When UDP is blocked by a firewall and TCP is used as a fallback option, there is a greater variation in payload size, probably due to TCP bundling upon retransmission, but still a low average payload size compared to the maximum



(a) BzFlag hit limits (world units).

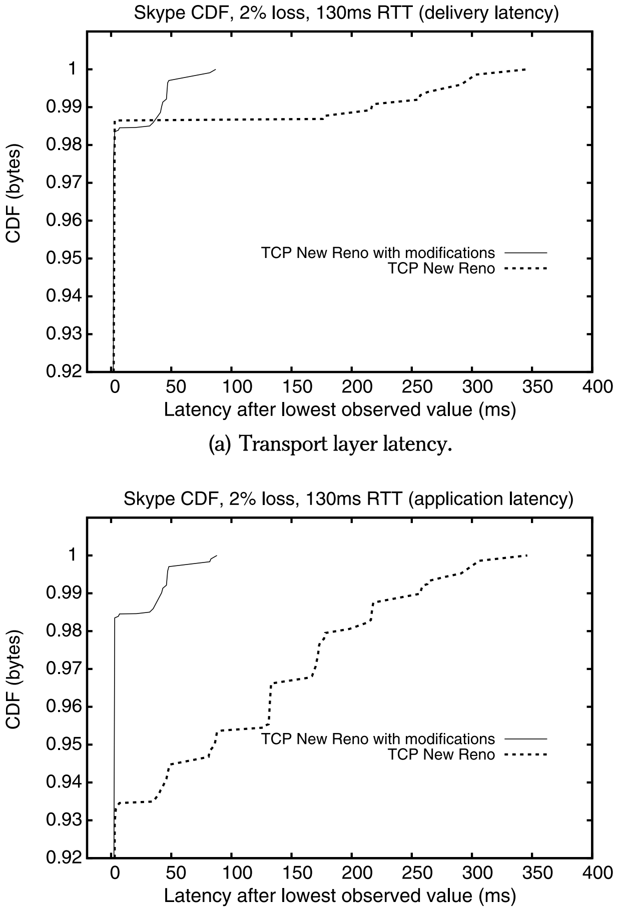
(b) BzFlag hit limits (per cent).

Figure 17.11 BzFlag hit limits for “perfect shot”.

segment size. The average interarrival time is approximately the same as for UDP. The result is that Skype over TCP requires slightly more bandwidth than Skype over UDP.

We wanted to investigate whether our modifications could improve the perceived quality of such a VoIP session. We blocked UDP traffic, forcing Skype to fall back to TCP. Regarding speech in VoIP conferences, differences between each conversation can make it difficult to compare one session with another. To have directly comparable data and to reproduce results, we chose to use sound clips, which we played across the Skype session. We experimented with several different sound clips, both with respect to the numerical results gathered from packet traces and to the subjective tests that are described below. Three different sound clips were ultimately chosen for a user test. Each clip was played twice, one with TCP modifications and one with regular TCP. The sound clip was sent across the Skype connection and the resulting output was recorded at the receiver.

Latency results. Figure 17.12 shows the statistical results for one of the sound clips. The CDF graphs show the relative differences in latency between the two streams. Figure 17.12(a) shows when the data is delivered to the receiver machine (at the transport layer), and we can see that TCP with modifications delivers lost payload much earlier than standard TCP. At the application layer, the situation is worse.



(a) Transport layer latency.

(b) Application layer latency.

Figure 17.12 Skype latency.

Figure 17.12(b) shows the application layer latency, and we see an even larger gain when using TCP with the modifications. Close to seven per cent of the data sent with standard TCP had to wait for one or more retransmissions, while only around two per cent (the loss rate) of the data sent with the modifications was stuck in the receive buffer. With the average interarrival time of 34 ms for this stream (see table 17.1), most of the data segments affected by loss were delivered with the next packet. In addition, even though a retransmission occurred, the data are still delivered faster than with standard TCP, greatly improving the application latency in order to meet the guidelines for the one-way transmission time of 150 to 200 ms before users become dissatisfied [20].

Impact on user perception. When taking part in a phone conversation, one of the most important aspects is the sound quality. Distortion and similar artefacts will reduce QoE, making it more difficult to understand the speaker. We therefore made recordings of Skype conversations played over links with loss and delay and had a group of people evaluate the quality. The same sound clip was played twice, once with TCP modifications and once without.

In order to generate the same original sound for both runs, we played the first minute of the clip. We present results using one speech podcast and two songs. The podcast was chosen because speech is what Skype is designed for and thus the compression format should be tuned for it. The songs were chosen because it is easier to notice artefacts when you have a rhythm to relate to. A complicating factor is that Skype encoding sometimes distorts the sound when using TCP, even under perfect network conditions (no loss or delay). All the recordings would, however, be exposed to these irregularities, so the resulting recordings should be directly comparable. Unfortunately, it was not possible to investigate the cause of the distortion further since Skype is a proprietary application.

As a reference test, we played the same version of one sound clip sent over a modified connection twice. This was done to ensure that a “memory effect” does not overly influence the answers, for instance, after listening to two clips, the listener prefers the first clip because he/she has already “forgotten” the faults observed therein.

All in all, we collected 88 votes and the results are shown in figure 17.13. The recordings made with the modifications were clearly preferred by the users. We were told that the differences in clip 1 (figure 17.13), which was the podcast, were small but still noticeable. With clip 3, which was one of the songs, the users commented that the version without the modifications was distinctly suffering in quality compared to the clip run using modified TCP. The test subjects complained about delays, noise, gaps, and others artefacts, and said that it was easy to hear the difference.

In the reference test (clip 2 in figure 17.13), the majority of the test subjects answered that they considered the quality as equal. Of the users who decided on one of the versions of this clip, most of them chose the one that was played first. This may be due to the memory effect (discussed above), where the listener may have forgotten the errors of the first clip. For clip 1, the modified version was the second clip to be played. The memory effect here may have diminished the results for the modified version of TCP. Even so, a great majority of the test subjects preferred the modified version. For clip 3 (figure 17.13), the order was the opposite (modified TCP first).

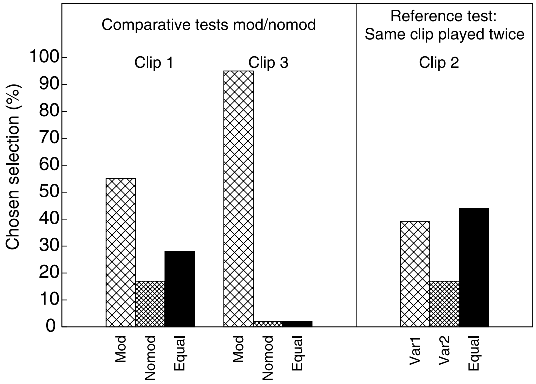


Figure 17.13 Preferred sound clips from Skype user tests.

We can assume that some of the people who chose the modified TCP version were fooled by the memory effect. The majority of subjects who chose the modified TCP version, however, is so great (95.4 per cent) that we can safely trust the numbers.

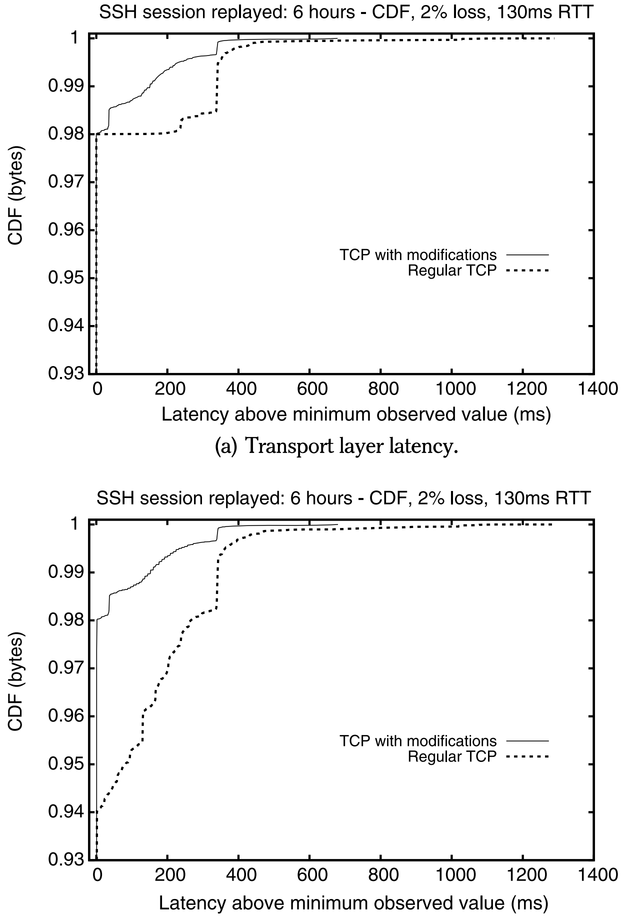
Remote Terminals: SSH

A very basic way of working on a remote computer is the common secure shell (SSH) protocol. It creates an encrypted connection to a remote computer, allows remote interaction, and exhibits clear thin-stream properties. While in an SSH session, loss can lead to delays that make the editing of text more difficult, e.g., lag between pressing keys and the subsequent screen update, and really annoy the user. We therefore analysed the latency for this application and determined that there is a noticeable difference between using regular TCP and the modified version.

Latency results. Figure 17.14 shows the CDFs that resulted from analysing SSH packet traces. A captured SSH session was replayed over the test network for six hours. The resulting dumps were analysed with respect to transport and application layer latency. The SSH session statistics (see table 17.1) show that the average interarrival time is high and the packets are small. This means that all of the modifications are active. For the Skype test, the interarrival time was relatively low, so RDB was the mechanism that contributed most. With the high interarrival times for the SSH test, the retransmission mechanisms were more in demand. We can see from figure 17.14 that the share of data that experience undue delays was also smaller for this stream when using the modifications. Figure 17.14(a) shows that 98 per cent of the data was delivered with no extra delay, reflecting the loss rate of two per cent. When loss was experienced, the delivery latency for regular TCP severely increased compared to the modified version.

When comparing the transport layer latency (figure 17.14(a)) to the application layer latency (figure 17.14(b)), we can see that waiting for lost segments is a major delaying factor for as many as six per cent of the packets for regular TCP. The high maximum values that we observed were due to the high average interarrival time for this stream. This resulted in more retransmissions by timeout, and the potential triggering of exponential backoff. For the unmodified TCP test, most of the lost segments were recovered after approximately 330 ms, indicating that retransmission by timeout was the prime mechanism of recovery. For the modified version, a more spread-out pattern reflects the triggering of the modifications. As for the Skype test, we observe that the delivery latency for the application-layer over modified TCP is almost identical to the transport layer latency.

Impact on user perception The QoE of using a remote text terminal can be severely diminished by network loss. The screen may not be updated with the character that was typed, and it may be difficult to edit the document in a controlled manner. After



(a) Transport layer latency.

(b) Application layer latency.

Figure 17.14 SSH latency.

analysing the SSH latency, we wanted to test if the improvements in application layer latency could be noticed by the user.

In the test, the users opened a command window on one computer and initiated a text-based SSH connection to the receiver. Each user then opened a text editor (like vi or emacs) and typed a few sentences. The users were encouraged to try to keep their eyes on the screen while typing in order to observe any irregularities that might occur. In order to make the test applicable to the users that prefer watching the keyboard while writing, a second test was devised. This test consisted of repeatedly hitting the same key while watching the screen. After the typing, the user was to close the editor and log out of the SSH session. We then made a change in the network configuration and the user repeated the procedure. Afterwards, the test subjects had to decide which of the two sessions they considered to have the best performance. In order to avoid the memory effect, half of the test subjects took the modified TCP test first and the other half took the regular TCP test first. A total of 26 people participated in this test. All of the participants were familiar with text editing over SSH and used a text editor that they were comfortable with.

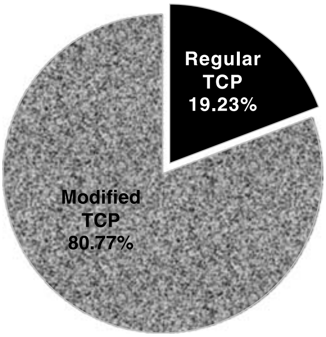


Figure 17.15 SSH user test: Preferred connection.

Figure 17.15 shows how many chose each type of session. The black area represents the per centage of the test subjects who preferred the regular TCP session (19 per cent), while the hatched area represents the ones who preferred the session that used the modified TCP (81 per cent). A frequently occurring comment from the users was that they preferred the connection using the modified TCP because the disturbances that occurred seemed less bursty. That is, if they had written a group of letters, they showed up without a large delay. Another recurring comment was that deleting became more difficult in the sessions using regular TCP because it was easy to delete too many letters at a time.

The large share of users who preferred the session using the modified TCP strongly suggests that the improvement shown in figure 17.14 can also be experienced at the user level. The number of participants for the user tests may statistically, be too small to draw absolute conclusions. Seen in correlation with the measured data, however, we feel that this is a strong indication of the impact of the TCP modifications on user experience for the thin-stream applications tested.

Discussion

The thin-stream TCP modifications are based on the concept that the applications in question never use their fair share of the bandwidth. The price paid for the reduced latency is a greater measure of redundant transmissions.

For the exponential backoff and dupACK modifications, the price comes in the form of more aggressive retransmissions. In a draft[6] to the Internet Engineering Task Force, Allman et al. suggest that measures should be taken to recover lost segments when there are too few unacknowledged packets to trigger fast retransmit. They propose early retransmit, which reduces waiting times for any of four situations: The congestion window is still initially small; it is small because of heavy loss; flow control limits the send window size; or the application has no data to send. The draft proposes to act as follows whenever the number of outstanding segments is lower than four: If new data are available, they follow limited transmit [15]; if none are available, this reduces the number of duplicate packets necessary to trigger fast retransmit to as low as one depending on the number of unacknowledged segments.

Our fast retransmit-triggering mechanism has no stepwise escalation, but is fully applied when there are few packets in flight. This is because we can expect the thin stream to keep its properties throughout its lifetime and also because so many thin-stream applications are interactive with strict latency requirements. Allman et al. try to prevent retransmission timeouts by retransmitting more aggressively, thus keeping the congestion window open. If their limiting conditions change, they still have higher sending rates available. Our main goal is to keep the delivery latency low. We have no motivation to prevent retransmission timeouts in order to keep the congestion window open and retransmit early to reduce application-layer latencies. The removal of the exponential backoff can of course result in spurious retransmissions when the RTT changes. The proposed method of TCP Santa Cruz [12] uses TCP timestamps and TCP options to determine the copy of a segment that an acknowledgement belongs to and can therefore provide a better RTT estimate. Since the RTT estimate can distinguish multiple packet losses and sudden increases in actual RTT, TCP Santa Cruz can avoid exponential backoff. The ability of Santa Cruz to consider every ACK in RTT estimation has minor effects in our scenario, where few packets are generated. The ability to discover the copy of a packet to which an ACK refers would still be desirable but would require receiver-side changes that we avoid.

For *the RDB modifications*, there is a constant factor of redundancy giving larger but not more packets. The increase is dependent on the RTT and interarrival times; that is, connections experiencing a high RTT and low interarrival times will have a high level of redundancy and an increased number of spurious retransmissions. Several packets will be sent between each ACK and thus bundles can be made. The increase in bandwidth, however, is still small due to the fact that the applications only

[6] IETF Draft draft-allman-tcp-early-rexmt-07: Mark Allman, Konstantin Avrachenkov, Urtzi Ayesta, Josh Blanton, and Per Hurtig, “Early Retransmit for TCP and SCTP”, June 2008, expired December 2008.

transmit sporadically (or at a steady low rate). Currently RDB bundles as much data as possible into one packet, only limited by a specified byte threshold or the network maximum transfer unit (MTU) size. Reducing the limit for the size of bundled packets would help to avoid the bundling of too many segments on, for instance, a high RTT connection. An aspect that needs further study is that RDB can, in principle, affect congestion control by hiding packet loss and thus avoiding a decrease in congestion window size. In reality, this situation will appear only under very special conditions, as a the TCP implementation fills packets to their maximum segment size when streams are greedy senders.

Before our thin-stream investigation is finally concluded, we need to design and perform experiments (or simulations) to determine whether the modified TCP is fair in borderline cases or how that can be achieved. We would also like to see if our TCP enhancements could be included into the Linux distribution. We have already sent the kernel patch to an international bank whose IT director had publicly asked for solutions to TCP's latency problems. With his encouragement and that of the Linux distributors, we initiated discussions on the Linux mailing lists after a presentation at the Linux-Kongress in October 2008.

Multicast Group Communication

High latency does not only occur because of the shortcomings of TCP. It is, of course, also strongly affected by the physical distance between users. A means of reducing this is to exploit group communication. The users of many interactive applications with a large number of concurrent users can be grouped together. In VoIP systems, these groups are formed by individual conversations of audio conferences and, in games, these groups are formed by areas of interest. This grouping can be used for latency reduction. As the geographical dispersion of users of interactive services depends heavily on the time of day, it is possible to relocate server functions to local proxy servers and reduce the average latency of those accessing it, or use a peer-to-peer-like approach for exchanging state information. Once this approach is supported, decisions can be made for each group separately. We have used graph algorithms that construct overlay multicast trees where we assume that a central entity, such as a game server of a conference mixer, manages and sets up the group communication, but the communication itself is a graph connecting the users in an overlay network.

Building Efficient Group Communication Structures

Interactive applications come with individual constraints that must be met to achieve a desired QoE, but not only that, the constraints can vary within a single application. An application may therefore make use of several group communication strategies at once, which addresses one aspect of the application's communication. For each aspect, an own overlay network can be built, where graph-theoretical results are the basis for the formation of each of these overlays. The graphs that are considered consist of nodes, where the set of nodes comprises all computers that participate in the application, and edges, which are formed by the Internet paths between every pair of nodes. The latter is noteworthy because it means that the graphs at the basis of group communication considerations are fully meshed (complete). Application group definitions, such areas of interest or conversations, define subgraphs within this complete graph. These subgraphs are also fully meshed, but this does not preclude the use of other nodes of the entire graph if that can improve communication performance.

We have therefore evaluated the different approaches against different metrics such as *diameter* (maximum pair-wise latency between any two nodes), *average pairwise latency*, *stress* (bandwidth consumed at each node), *execution time* (time it takes to find the suitable tree), *total cost* (sum of all connection latencies), and *stability* (number of paths that are modified in an update). Based on section 17.3, we know also that some metrics do not need to be minimised, but that they are bounded instead. This is the diameter in particular, but could also be the case for stress. We have investigated three main ways of constructing multicast trees: spanning tree algorithms [34, 19], Steiner tree algorithms [34, 7], and dynamic tree algorithms [4].

Full Reconfiguration Algorithms

Algorithms that regenerate graphs from scratch every time a node joins or leaves include spanning tree and Steiner tree algorithms. *Spanning-tree* algorithms build a spanning tree on an input graph, where a spanning tree is a connected acyclic subgraph (tree) that connects all the nodes [30]. Here, we have compared and developed a large number of enhanced algorithms based on known algorithms like minimum spanning trees (MSTs) and shortest-path trees (SPTs). Then, to optimise towards given goals, for example, minimum diameter (md), or to adhere to specified bounds, for example, degree limits (dl) or bounded diameter (bd), we modified the algorithms. Furthermore, *Steiner-tree* algorithms can include nodes into a tree that are willing to forward traffic although they are not receivers themselves. They may be added to perform better with respect to the target metrics. We used algorithm ideas from the well-known minimum-cost Steiner-tree heuristics shortest-path heuristic (SPH) [5], distance-network heuristic (DNH) [2] and average-distance heuristic (ADH) [10]. Based on these algorithms, we devised Steiner-tree heuristics with additional bounds. An overview of these algorithms is given in table 17.2 [7].

A typical evaluation of these algorithms can be seen in figure 17.16, which shows plots of the diameter of trees using degree-limited shortest path tree heuristic (dl-SPT) and dl-SPTs with Steiner points (sdl-SPT) with various degree limits. The graph shows that heuristics based on degree-bounded (i.e., stress-limited) Dijkstra's shortest path can achieve an improvement in diameter when they can avail themselves of Steiner points.

Dynamic Algorithms

Many interactive applications have group dynamics where users join and leave a group dynamically, e.g., leaving one room and entering another in a virtual environment. In such a case, spanning and Steiner tree algorithms are too costly when execution time or the stability of trees matters. Thus, we have also looked at dynamic algorithms that reconfigure only local parts of a graph [4]. This work has resulted in

Algorithm	*Meaning*	*Optimize*	*Algorithm basics*	*Steiner points*	*Constraint*	*Complex.*	*Ref.*
SPH	Shortest Path heuristic	total cost	Prim's MST	✓	-	$O(pn^2)$	[5]
DNH	Distance network heuristic	total cost	Prim's MST	✓	-	$O(pn^2)$	[2]
ADH	Average distance heuristic	total cost	Kruskal's MST	✓	-	$O(n^3)$	[10]
dl-SPH	Degree limited SPH	total cost	Prim's MST	✓	degree	$O(pn^2)$	[7]
dl-DNH	Degree limited DNH	total cost	Prim's MST	✓	degree	$O(pn^2)$	[7]
md-SPH	Minimum diameter SPH	diameter	md-OTTC	✓	-	$O(n^3)$	[7]
md-DNH	Minimum diameter DNH	diameter	md-OTTC	✓	-	$O(n^3)$	[7]
bdo-SPH	Bounded diameter optimized SPH	total cost	OTTC	✓	diameter	$O(n^3)$	[7]
bdo-DNH	Bounded diameter optimized DNH	total cost	OTTC	✓	diameter	$O(n^3)$	[7]
bdr-SPH	Bounded diameter randomized SPH	total cost	RGH	✓	diameter	$O(pn^2)$	[7]
bdr-DNH	Bounded diameter randomized DNH	total cost	RGH	✓	diameter	$O(pn^2)$	[7]
mddl-SPH	Minimum diameter degree-limited SPH	diameter	mddl-OTTC	✓	degree	$O(n^3)$	[7]
mddl-DNH	Minimum diameter degree-limited DNH	diameter	mddl-OTTC	✓	degree	$O(n^3)$	[7]
smddl-OTTC	Steiner minimum diameter degree-limited OTTC	diameter	Prim's MST	×	degree	$O(n^3)$	[19]
bddlo-SPH	Bounded diameter degree-limited optimized SPH	total cost	dl-OTTC	✓	diam./degree	$O(n^3)$	[7]
bddlo-DNH	Bounded diameter degree-limited optimized DNH	total cost	dl-OTTC	✓	diam./degree	$O(n^3)$	[7]
bddlr-SPH	Bounded diameter degree-limited randomized SPH	total cost	dl-RGH	✓	diam./degree	$O(pn^2)$	[7]
bddlr-DNH	Bounded diameter degree-limited randomized DNH	total cost	dl-RGH	✓	diam./degree	$O(n^3)$	[7]
sdl-OTTC	Steiner degree-limited OTTC	total cost	Prim's MST	×	diam./degree	$O(n^3)$	[**?**]
sdl-RGH	Steiner degree-limited RGH	total cost	Prim's MST	×	diam./degree	$O(n^2)$	[**?**]
s-SPT	Steiner Dijkstra's SPT	src eccentr.	Dijkstra's SPT	×	-	$O(n^2)$	[**?**]
br-SPH	Bounded radius SPH	total cost	Prim's MST	✓	radius	$O(n^3)$	[7]
br-DNH	Bounded radius DNH	total cost	Prim's MST	✓	radius	$O(n^3)$	[7]
dl-SPT	Degree-limited Dijkstra's SPT	src eccentr.	Dijkstra's SPT	×	degree	$O(n^2)$	[**?**]
sdl-SPT	Steiner degree-limited Dijkstra's SPT	src eccentr.	Dijkstra's SPT	✓	degree	$O(n^2)$	[**?**]
brdl-SPH	Bounded radius degree-limited SPH	total cost	Prim's MST	✓	radius/degree	$O(n^3)$	[7]
brdl-DNH	Bounded radius degree-limited DNH	total cost	Prim's MST	✓	radius/degree	$O(n^3)$	[7]

Table 17.2 Examples of Steiner tree heuristics evaluated.

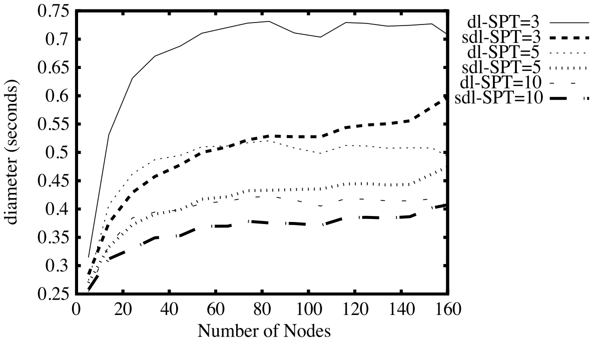


Figure 17.16 Diameter (seconds) of sdl-SPT and dl-SPT with different degree limits depending on group size [7].

a number of insert and remove algorithms that can be applied in pairs, have very short execution times, and achieve similar results.

Reducing Overlay Construction Time

As an alternative to the separate join and leave heuristics that are applied in pairs, the execution time of spanning and Steiner tree heuristics can also be reduced by pruning the fully meshed input graph instead. While this strategy cannot in general guarantee the stability of the resulting graphs, it can nonetheless be realistic. Many commercial interactive applications are supported by infrastructure components such as proxy servers, and we investigated several pruning strategies that concentrate links in such proxies [19, 31].

Achieving Up-to-Date Network Information

In the work described above, we found many usable algorithms for several different scenarios with different optimisation goals. A general challenge, however, is that these algorithms have an underlying assumption of full network knowledge. To create and maintain an efficient group communication structure, we need up-to-date network information about load, latencies, and bandwidth. Achieving full, up-to-date knowledge of the network requires monitoring and is nearly impossible for a large number of nodes because the monitoring traffic grows quadratically with the number of nodes. This scalability problem is addressed by techniques that estimate link latencies and costs, but the trade-off is their accuracy. We have therefore compared tree construction algorithms executing with limited knowledge with the same algorithms having full topological knowledge [36].

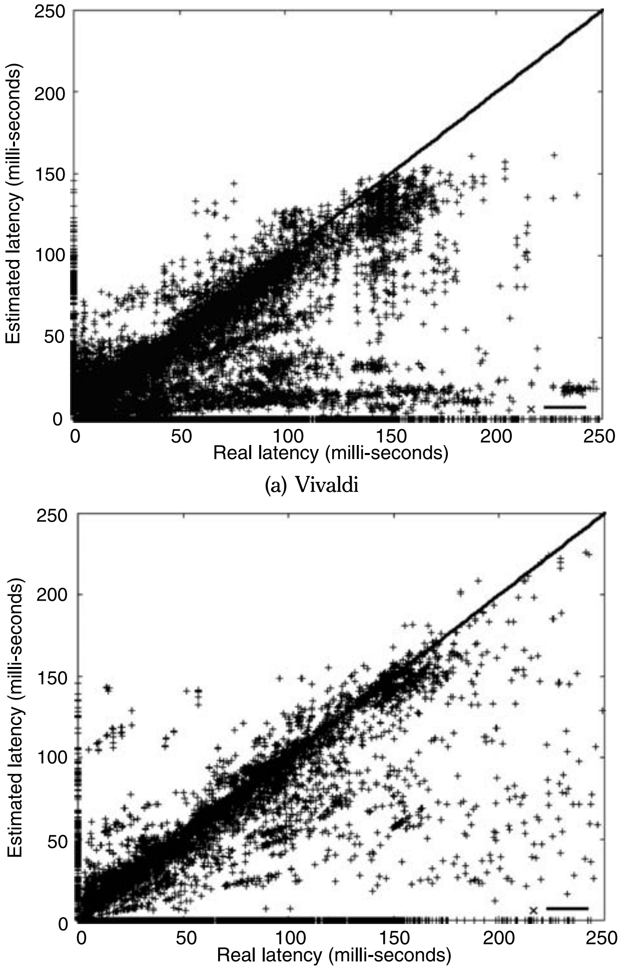
We implemented a group communication system that makes use of the link estimation techniques *Vivaldi* [1] (which includes latency-probing information in each packet) and *Netvigator* [17] (which uses known landmark nodes), and we compared their estimated latencies with all-to-all measurements using *ping*. We performed estimation experiments repeatedly over a ten-day period and included 215 PlanetLab nodes; the total number of nodes we were allowed to access.

Figure 17.17 visualises the discrepancy between the measured ping latencies and the estimated latencies in greater detail. The plots are scatter plots of all pairs of nodes that compare the measured RTT with the estimated RTT. The plots show that Netvigator is very accurate in its estimations, whereas Vivaldi has a bit more variation. Vivaldi, and to some degree Netvigator, also overestimates RTTs for the smaller actual RTTs, while it underestimates longer distances. When the actual RTTs are very small, the overestimations are relatively high, but the absolute deviation may still be acceptable for many applications.

This alone, however, does not determine whether group communication trees are constructed appropriately based on these estimations. To test this, we built trees using dl-SPT, which aims at low tree diameters [19]. In figure 17.18, we see the discrepancy between the diameter that is achieved when the heuristic relies on the estimations and the diameter that is achieved when it has full knowledge of the network. Clearly, there is an increase in diameter, and Netvigator performs better than Vivaldi. However, we found that both estimation techniques will work well enough in many interactive applications, including conferencing and games. Vivaldi recovers better from membership dynamics and is very easy to deploy in a peer-to-peer fashion. Netvigator needs an infrastructure, which may be available when an infrastructure of proxy servers exists.

17.4 Conclusion and Future Perspectives

The RELAY members are interested in distributed system performance and resource utilisation, especially in the area of multimedia. Activities range from low-level operating system enhancements to high-level architectures and application mechanisms. We have here focused on the results from promising system enhancements for in-

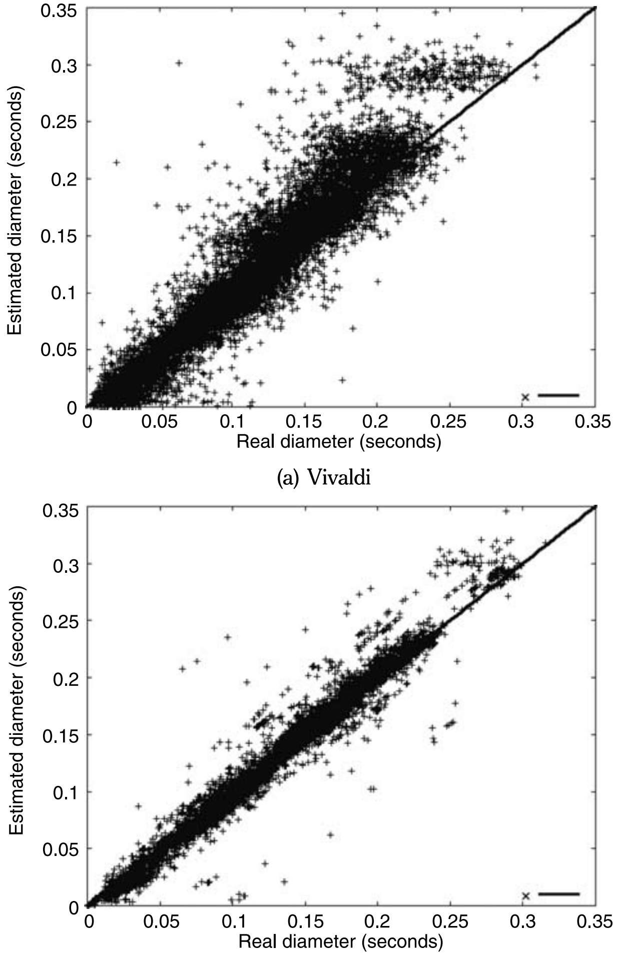


(b) Netvigator

Figure 17.17 Scatterplot of measured versus estimated latency.

teractive traffic which requires low latency to achieve a good QoE. We also have several other activities, however, such as, in particular, multiprocessor scheduling and video streaming.

In the next few years, RELAY will continue research on performance and resource utilisation by designing, implementing, enhancing, integrating and evaluating mechanisms, algorithms, and tools to improve resource utilisation, increase throughput, reduce latency, improve user QoE, and support soft QoS. Our target scenario will continue to be distributed multimedia systems, but we will evaluate new, emerging applications, systems, and techniquesIn particular, the current long-term strategy



(a) Vivaldi

(b) Netvigator

Figure 17.18 Scatter plot comparing diameters achieved by dl-SPT based on measured and estimated latency.

aims at all research topics that are pursued either in the Centre for Research and Innovation *iAD* [35], the StorIKT project *Verdione* [3], or both. There are several links between these two projects, but iAD is generally targeted towards the performance of future search and data delivery systems, and Verdione targets system support for the World Opera scenario with distributed (opera) performances. While these application scenarios are quite different, underlying system support may be shared. Thus, these two subprojects are umbrellas for the smaller activities, which puts all of them into a larger context of RELAY, with visions for optimised, resource-efficient distributed multimedia systems that support applications ranging from low-latency, interactive games to high bandwidth streaming of multiple video–and everything in between.

Acknowledgments

Thank you to *all* the members of the RELAY group, including every master student, doctoral student and postdoc. You have all given us new information and insights and presented valuable results to the research in RELAY.

References

[1] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. *ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 15–26, 2004.

[2] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15:141–145, jun 1981.

[3] The Verdione project. Verdione. https://verdione.org/.

[4] K.-H. Vik, C. Griwodz, and P. Halvorsen. Dynamic group membership management for distributed interactive applications. *IEEE Conference on Local Computer Networks (LCN)*, pages 141–148, oct 2007.

[5] H. Takahashi and A. Matsuyama. An approximate solution for the steiner trees in graphs. *Intl. J. Math. Japonica*, 6(1):573–577, 1980.

[6] Skype, March 2008.

[7] K.-H. Vik, P. Halvorsen, and C. Griwodz. Evaluating steiner tree heuristics and diameter variations for application layer multicast. *Elsevier Computer Networks*, 52(15):2872–2893, oct 2008.

[8] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control . RFC 2581 (Proposed Standard), Apr. 1999.

[9] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001 (Proposed Standard), Jan. 1997.

[10] V. Rayward-Smith and A. Clare. The computation of nearly minimal Steiner trees in graphs. *International Journal of Mathematical Education in Science and Technology*, 14(1):8pp, 1983.

[11] W. chang Feng, F. Chang, W. chi Feng, and J. Walpole. A traffic characterization of popular on-line games. *IEEE/ACM Transactions on Networking*, 13(3):488–500, 2005.

[12] C. Parsa and J. J. Garcia-Luna-Aceves. Improving TCP congestion control over internets with heterogeneous transmission media. *International Conference on Network Protocols (ICNP)*, pages 213–221, Nov. 1999.

[13] R. J. Larsen and M. L. Marx. *An Introduction to Mathematical Statistics and Its Applications*. Prentice Hall, 1986.

[14] W. chang Feng, F. Chang, W. chi Feng, and J. Walpole. Provisioning on-line games: a traffic analysis of a busy Counter-strike server. *In the Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, Marseille, France,* pages 151–156, November 2002.

[15] M. Allman, H. Balakrishnan, and S. Floyd. Enhancing TCP's Loss Recovery Using Limited Transmit. RFC 3042 (Proposed Standard), Jan. 2001.

[16] M. Claypool. The effect of latency on user performance in real-time strategy games. *Elsevier Computer Networks,* 49(1):52–70, Sept. 2005.

[17] P. Sharma, Z. Xu, S. Banerjee, and S.-J. Lee. Estimating network proximity and latency. *SIGCOMM Comput. Commun. Rev.*, 36(3):39–50, 2006.

[18] The MiSMoSS project. Middleware services for management of shared state in large-scale distributed interactive applications (MiSMoSS). http://www.simula.no/research/networks/projects/RELAY/mismoss/.

[19] K.-H. Vik, P. Halvorsen, and C. Griwodz. Multicast tree diameter for dynamic distributed interactive applications. *infocom,* pages 1597–1605, Apr. 2008.

[20] International Telecommunication Union (ITU-T). One-way Transmission Time, ITU-T Recommendation G. 114, 2003.

[21] R. A. Bangun, E. Dutkiewicz, and G. J. Anido. An analysis of multi-player network games traffic. *Proceedings of IEEE MMSP,* 1999.

[22] M. S. Borella. Source models of network game traffic. *Proceedings of Networld and Interop,* 1999.

[23] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. An empirical evaluation of TCP performance in online games. *Proceedings of ACM SIGCHI ACE'06,* Los Angeles, USA, Jun 2006.

[24] P. Svoboda, W. Karner, and M. Rupp. Traffic analysis and modeling for world of warcraft. *Communications, 2007. ICC '07. IEEE International Conference on,* pages 1612–1617, June 2007.

[25] K.-T. Chen, P. Huang, C.-Y. Huang, and C.-L. Lei. Games traffic analysis: An MMORPG perspective. *nossdav,* pages 19–24. ACM Press, 2005.

[26] The RELAY project. Resource utilization in time-dependent large-scale distributed systems (RELAY). http://www.simula.no/research/networks/projects/RELAY.

[27] C. Griwodz and P. Halvorsen. The fun of using TCP for an MMORPG. *nossdav,* pages 1–7. ACM Press, May 2006.

[28] Z. Wang. *Internet QoS: Architectures and Mechanisms for Quality of Service.* Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2001.

[29] M. Claypool and K. Claypool. Latency and player actions in online games. *com_acm,* 49(11):40–45, Nov. 2005.

[30] B. Y. Wu and K.-M. Chao. *Spanning Trees and Opitmization Problems.* Chapman and Hall/CRC, 2004.

[31] K.-H. Vik, P. Halvorsen, and C. Griwodz. Constructing low-latency overlay networks: Tree vs. mesh algorithms. *IEEE Conference on Local Computer Networks (LCN),* Oct. 2008.

[32] L. Pantel and L. Wolf. On the impact of delay on real-time multiplayer games. *nossdav,* pages 23–29, 2002.

[33] K. Evensen, A. Petlund, C. Griwodz, and P. Halvorsen. Redundant bundling in tcp to reduce perceived latency for time-dependent thin streams. *Communications Letters, IEEE*, 12(4):324–326, April 2008.

[34] K.-H. Vik, C. Griwodz, and P. Halvorsen. Applicability of group communication for increased scalability in MMOGs. *Workshop on Network and System Support for Games (NETGAMES)*, Singapore, Oct. 2006. ACM Press.

[35] The iAD project. information access disruptions (iAD). http://www.ifi.uio.no/forskning/grupper/nd/projects/2006/iad/.

[36] K.-H. Vik, C. Griwodz, and P. Halvorsen. On the influence of latency estimation on dynamic group communication using overlays, *(to appear)*. *SPIE/ACM Conference on Multimedia Computing and Networking (MMCN)*, San Jose, CA, USA, Jan. 2009.

18

SCIENTIFIC COMPUTING — WHY, WHAT, HOW AND WHAT'S NEXT

Hans Petter Langtangen and Joakim Sundnes

Why Scientific Computing is Important

Problems in science and engineering have traditionally been solved by a combination of theory and experiment. In many branches of science, the theories are based on mathematical models, usually in the form of equations describing the physical world. By formulating and solving these equations, one can understand and predict the physical world. The theories are constructed from or validated by physical experiments under controlled conditions.

A fundamental problem with equation-based theories is that the equations are extremely hard to solve by mathematics, pen, and paper. Only a fraction of the interesting ones, often in an oversimplified form, can be attacked by the mathematical techniques developed over the centuries. Despite this difficulty in solving equations, the combination of theory and experiment has been tremendously successful in advancing technology and thereby increasing the living standard of human beings. Cultural life is also advanced through our deeper insight into how nature works.

With computers and the new mathematical methods developed for them, it is now possible to solve large classes of equations that occur in scientific theories. This fact has made the theories far more applicable, since computers can be used as a virtual

Hans Petter Langtangen · Joakim Sundnes
Simula Research Laboratory

Hans Petter Langtangen · Joakim Sundnes
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_18,

laboratory where processes in nature and technological devices can be simulated. Using simulations to gain scientific insight is often referred to as *computational science*. We are only in the beginning of the computational science age. Future progress in this area may greatly accelerate our understanding of nature and allow us to reach new technological levels we can hardly imagine today.

To exemplify the idea of computational science, think of developing a new airplane. In the pre-computer age, one had to build and test real airplanes, first at the model scale and then full scale, to judge their flight properties. Today, airplanes can be designed in a computer, the flow past the airplane can be computed, and from the flow one can determine the flight properties. With a computer model of the airplane it is easy to change the design and determine its influence on the flight properties. Computer experiments of this kind have already led to increased safety and significant reduction in fuel consumption.

This computational approach is now adopted throughout science and engineering: in weather forecasts, climate predictions, car design, oil recovery, hydropower, computer chip design, and tsunami warning systems, to mention just a few well-developed application areas. In certain areas of medicine and geoscience, mathematically based theories and their associated computations are just starting to receive significant attention, and Simula's Scientific Computing group has been active in these new, exciting application areas. Equations very similar to those governing an airplane's flight can be used to investigate the effect of surgery on blood flow, for example, when replacing a heart valve or performing a bypass operation.

Using the computer as a laboratory requires a rich collection of computational techniques. The scientific computing *discipline* aims at developing such techniques, including algorithms for solving mathematical problems arising from scientific theories, analysis of the behaviour of such algorithms, computer implementation of the algorithms, the composition of different algorithms to solve complex mathematical problems, and the development of associated, complex software systems to perform simulations. Techniques from scientific computing are indispensable to computational science.

Some of the most common type of equations entering into scientific theories are called *partial differential equations*. The unknowns to be computed from such equations are often functions of space and time and can describe physical quantities such as temperature, pressure, fluid velocity, solid displacement, electric current, and magnetic field. Algorithms and software tools for solving partial differential equations are particularly important areas of scientific computing, and form the main research topics in the Scientific Computing group at Simula.

How Scientific Computing is Conducted at Simula

The Scientific Computing group at Simula has its roots back in the early 1990's, first at the research institution SINTEF and later at the University of Oslo. The research goal has been to attack challenging problems described by partial differential equations, develop appropriate methods to solve them and critically investigate the methods' performance and reliability, either theoretically or by computer experiments. The problems originated from a diversity of fields: hydrocarbon recovery, material technology, coastal engineering, financial engineering, and medicine. The research results are manifest in a considerable amount of journal papers and conference contributions. We have, however, also focussed on producing software and books. The background for this rather unusual focus is described next.

Like most other groups in scientific computing, we found that developing the software we needed for the research consumed a large portion of our resources. This is due to the very nature of programming: Tracking down errors in mathematical software and providing evidence of the results' reliability are known to be challenging tasks. It was therefore decided, already in 1991, to develop a programming framework that could enhance our software development productivity. We named the framework *Diffpack* [2, 4, 8] and set the goal of distributing it worldwide. A huge amount of work was required to develop Diffpack, which was made possible by substantial funding from the Research Council of Norway in the period 1991–1996. All our research in scientific computing was implemented in Diffpack, such that scientists and engineers could more easily access our results through working in Diffpack code. This approach clearly increased the impact of our research in scientific computing, and has been a prevailing idea in our research group up to the time of this writing.

When Simula was founded in 2001, the Scientific Computing group had a well-established research profile with a primary focus on developing algorithms and associated Diffpack modules for solving partial differential equations in a variety of applications from physics, chemistry, medicine, and finance. The variety of applications naturally arose because of funding, which since the late 1990's came in small portions from a variety of sources with correspondingly different interests. Larger budgets at Simula gave the group a unique possibility to break the common tradition of many small projects and form a few larger, focussed, long-term projects. By concentrating efforts and resources, as was done a decade before in the early stages of the Diffpack project, the hope was to create research results of higher quality, interest, and impact. Time has, since then, shown us the importance of such a focus.

In 2002, we reorganised our research into two main projects, one on simulating the electrical activity in the heart [9, 5] and the other continuing the development of the Diffpack software for partial differential equations [4]. Simulations of the heart became a major driving force for future software development but we also addressed other challenging application areas, such as tsunami simulations, to maintain the generic nature of Diffpack.

The project on simulating the electrical activity in the heart later broadened its focus with two related subjects. One of these has from 2004 to 2009 been funded by an Outstanding Young Investigator grant from the Research Council of Norway, and concerns the interaction of electrical signals with the mechanical forces and deformations in the heart muscle. The other research subject aims at diagnosing myocardial ischemia, a reversible precursor of heart infarction. Instead of employing the standard methods of contemporary medical research, we apply biophysically based mathematical models for analysing electrocardiogram (ECG) recordings and thereby identify whether a patien suffers from this disease [11].

Diffpack was commercialised in 1997 through the Norwegian company Numerical Objects and the software was no longer publicly available. We later realised that our research would gain more impact if it were based on an open-source platform. Some progress was made with an open-source version of Diffpack but not after the technology was sold to the German company inuTech in 2003. Therefore, in 2004 we launched a new initiative: Python Computing Components (PyCC). At this time, the group has substantial experience with Python and regards it to be a very promising programming platform for scientific applications [3, 6, 7]. The idea was to explore this platform and make a proof-of-concept study for a future software platform for partial differential equations, building on 13 years of experience with developing and distributing Diffpack. The development was highly application driven, with a primary focus on simulating the electrical activity in the heart.

Later, Simula became heavily involved in the international FEniCS project [1], which has the goal of producing the next generation of finite element software, brought forward by new programming techniques. The functionality of PyCC was merged with FEniCS and FEniCS now acts as the primary software platform for Simula's scientific computing activities. Nevertheless, a range of other packages are also used and FEniCS itself builds on numerous libraries. By focusing our research around a collection of open-source modules such as FEniCS, we enable other researchers around the world to immediately access the working code implementing the results in our papers and books. According to our experience with Diffpack, such an outreach of results is a key factor in gaining impact.

Around 2005 we undertook two major initiatives: an industrial collaboration with the oil company Hydro and a proposal for a Centre of Excellence. Both projects, being of significant size and duration, aimed at research of high quality and impact. Earning a Centre of Excellence in Norway requires winning a tight competition: Only eight out of 98 applications were successful in the 2005 call. Fortunately, our proposal was one of them. Our key ideas for this centre were to further develop our strong position in numerical algorithms and software for partial differential equations, but target a new and exciting application area, namely, blood and air flows in the human body. Such flows are known to be particularly difficult to compute, since they often interact with deforming soft tissues. In addition, there are occasionally local turbulences, especially in airway flow. Successful handling of these topics requires us to bring together expertise from mathematics, numerical analysis, scientific software, fluid and solid mechanics, biophysics, and medicine. Conducting such multidisciplinary research is challenging, and initiating collaborations with other excellent groups in other fields of science demanded quite an effort during the centre's first years.

The collaboration with Hydro has grown steadily from 2005 to the time of this writing and is now, from a financial point of view, larger than the Centre of Excellence grant. Hydro merged with Statoil in 2006 to form an oil company, StatoilHydro, of significant international size. The overall goal of our project with StatoilHydro is to provide computational tools that can enhance the probability of locating oil and gas. StatoilHydro faces a range of new challenges when moving from the Norwegian continental shelf to many different sites worldwide. It also appears that computational tools for oil and gas *exploration* have been much less developed than those for oil and gas production. Therefore, the field of oil and gas exploration constitutes an exciting and rich topic for scientific computing research. Similar to our biomedical applications, the geoscience projects with StatoilHydro are multidisciplinary and require an interplay between mathematics, numerics, geometric modelling, visualisation, geoscience, and software.

Our collaboration with StatoilHydro involves both basic research and the development of high-quality, professional software tools to be actively used by geoscientists in the company. Basic research activities, aiming at new methods to be published in international journals and conferences, are organised as part of Simula's Basic Research unit, while the development of software tools is carried out by Simula's subsidiary Kalkulo. All activities are fully financed by StatoilHydro. The scientific topics fall into five categories: storage and visualisation of 3D spatial geodata in geological time (a kind of "Google Earth", with data in depth and time); interaction between plate tectonics and mantle convection; modelling of geological space-time events (referred to as *compound modelling* in chapter 40); sedimentation arising from huge underwater sand avalanches; and 3D rock-fluid-heat interactions during basin developments. The results from basic research goes into Kalkulo's software tools and the usage of these tools by StatoilHydro poses new basic research questions. This effect and the substantial amount of funding are direct results of a very efficient and fruitful cooperation between the people at Simula and at StatoilHydro.

The Centre of Excellence proposal implied three new projects: Computational Middleware, basically continuing our previous project on software for partial differential equations (PyCC/FEniCS); the Robust Flow Solver project, for algorithmic research and the development of software for solving the Navier-Stokes equations; and Biomedical Flows and Structures, targeting blood and air flow applications as well as fluid-structure (blood-tissue) interactions.

From before gaining the Centre of Excellence we had two projects related to heart simulations: computing the electrical activity in the heart and using such computations to locate ischemic regions in the heart based on ECG measurements. In addition, we had the Computational Geosciences project with StatoilHydro. It soon became apparent that there was a significant flow of people, ideas, methods, and software tools between these projects. Essentially, the biomedical and geoscience problems are about solving partially differential equations that describe highly complex, heterogeneous media with many interacting physical processes. From the points of view of mathematical modelling, numerical methods, and software development, the applications share many of the same challenges and call for technical solutions of the same nature. Therefore, it was natural to tightly integrate the projects. We decided to do so within the framework of the Centre of Excellence, which was named the *Center for Biomedical Computing* and now contains all the scientific computing activities at Simula, plus external collaborating groups, the foremost being biomechanics at the Norwegian University of Science and Technology in Trondheim, fluid mechanics at the Norwegian Defence Research Establishment outside Oslo, computational engineering at Cambridge University, and stochastic uncertainty modelling at Texas A&M University. Adding up all its financial contributions, the CBC now has three times as much as Simula's own funding of scientific computing activity.

The preceding paragraphs describe how we have conducted our research from an organisational point of view, that is, how various topics played together over time and formed a Centre of Excellence with differing applications but a large core of common mathematical, numerical, and programming methods. A more fundamental aspect of conducting research concerns the types of problems and results we seek. All our problems are applied, in the sense that there is a significant future need for solutions to these problems in society. Our primary type of result follows a strong scientific tradition, namely, publication in well-known, peer-reviewed archival journals. The impact of journal papers alone, however, can be limited in today's world. We already emphasised the importance of implementing theoretical results in running software to reach broader audiences and thereby bring more attention to our work. Another successful way of reaching broader audiences is to write books. We have over the years developed a close collaboration with Springer, which has resulted in a dozen books.

What Has Been Achieved

First, we briefly describe some of the most important research-based knowledge we have created. After that, we review some metrics related to the Scientific Computing group's work.

Research on simulating the electrical activity of the heart has been going on since the mid-1990's. It appears that the governing equations for the heart's electrical activity are hard to solve [16]. The system of equations contains both partial differential equations and ordinary differential equations. We have developed solvers for the ordinary differential equations that exploit their structure [12, 14] and result in an overall second-order accuracy in time for the heart simulator [15]. Using multilevel block preconditioning, we demonstrated both theoretically and through numerical experiments that the workload increases only linearly with the number of unknowns [13]. This is a very important and necessary result when conducting full-scale human computations with tens of millions of unknowns. Chapter 20 has more information on the mathematical models and computational methods used in simulations of the electrical activity of the heart.

Although the main contributions relate to numerical methods and software, we also studied arrhythmias and gained new insight into the link between cellular level properties and events at the macroscopic scale, such as an extra heart beat [17]. More recently this work was extended in a more analytical direction, and used to study fundamental arrhythmia mechanisms and for initial investigations of relevance to anti-arrhythmic drugs [18, 19].

Computing an ailing region in the heart based on body surface ECG measurements constitutes a very challenging medical, mathematical, and computational problem [10, 11]. Successful results in this project, however, can ultimately lead to a better ECG apparatus. The computational technology we developed provides an estimate for the position and size of the lesion/injury such that appropriate medical care can be performed. Efforts have been put into formulating appropriate, simplified mathematical models and proving them sufficiently stable for meaningful computations (sufficient stability is a major mathematical difficulty in this field). Currently, our computational methods are tested on a number of patients at Rikshospitalet. The results are promising but many challenging unexplored issues must be investigated in order to validate the system's potential benefits. Chapter 22 describes the project in more detail.

Simula and StatoilHydro have developed in close collaboration two novel technologies for the computer-based modelling of complicated geological structures on different spatial scales and over geological time. These technologies are of strategic importance to StatoilHydro. The lack of *reliable* simulations of how sediments are deposited to form new geological layers is a major weakness in current oil and gas exploration practices. Simula's recent contributions, however, to the calibration and quality assurance of depositional models are promising. Over the next couple of years we expect to reveal whether the developed techniques can fill this gap in the exploration workflow. On the more basic research side, a novel approach to simulating sand-water mixtures, based on particle and lattice Boltzmann models, is emerging. We have also overcome difficulties with scalability and high parallel performance of a multiphysics code with heterogeneities from real-world geological data. Quite naturally, there is a delay in publishing these results because the Simula staff had to invest significant efforts in learning more about the application area of exploration. More details about the Computational Geosciences collaboration with StatoilHydro are covered in chapter 40.

The new research projects arising from the Centre of Excellence proposal have been running for less than two years, which is a very short time period for producing significant results. Some activities, however, were started during the writing of the proposal and others build on previous software developments. Below we describe the most important achievements related to past and present software activities and some new promising collaborations with medical doctors.

The software distributed by the scientific computing group constitutes a major result that has made an important impact. Diffpack was first released with public access in 1995 and later commercialised. Diffpack's distribution resulted in hundreds of active users in a range of institution, including corporations such as DaimlerChrysler and Intel, the National Aeronautics and Space Administration, and universities such as Cambridge, Cornell, and Stanford. Diffpack's principal aim was to explore object-oriented programming through the C++ language in order to simplify the development of scientific software. When Diffpack was founded in 1991, this programming technology was in its infancy in high-performance computing applications and was naturally met with much scepticism. A decade later, the programming technology had become a worldwide de facto standard for new scientific software projects. We believe Diffpack has provided an important example of this technology and how it can be used to solve real-world problems in industry and science. Similar packages appeared during the 1990's and provided many further demonstrations of the programming technology.

The FEniCS software brings the Diffpack ideas a big leap forward, as explained in chapter 23. Although FEniCS has not yet been officially released, test versions have become popular in over 40 countries, with about 500 downloads per month. There are at present almost 20 very active developers and contributors. The current popularity of FEniCS is expected to increase as the software becomes more stable and an official release is advertised. It is already recognised as a major innovation in the way in which future scientific software should be constructed. The modules in FEniCS mostly use Python as the user programming platform, while the computationally heavy work is done in either general Fortran, C, or C++ libraries or automatically generated C++ code tailored to the user's input. Through books, papers, talks, tutorials, and software, we have demonstrated how and why Python is a very promising programming platform for computational science. This result in itself has received significant interest and attracted attention to Simula's research.

Parts of the PyCC software have received attention from the American company Insilicomed, which is in the process of integrating this software in a commercial medical application for a market of significant size.

Developing state-of-the-art scientific software of industrial quality within an academic institution is challenging. The bread and butter of the classic researcher is writing scientific publications and, as a consequence, developing software to mint condition might not be given a high enough priority. Our experience shows that producing high-quality software that is downloaded and used by thousands of users worldwide indeed requires a skilled technical staff that can provide support and facilities such as automatic testing frameworks and software building on a variety of Linux, Windows, and Mac systems.

Although biomedical flows constitute a new research topic at the Centre of Excellence, some important results have already been obtained. We have established very fruitful collaborations with medical doctors at the University Hospital in Northern Norway and the University of Wisconsin in Madison. Simulating blood flow in a vessel system called the circle of Willis, at the base of the brain, has helped medical doctors formulate and confirm hypotheses regarding the formation and rupture of aneurysms and the results have already been published in a pure medical journal. We have made progress with the Madison group in simulating the pulsatile flow of the cerebrospinal fluid in the upper spinal canal. Understanding this flow may be important for understanding Chiari I malformations, where parts of the brain have sunken down towards the neck, which may result in a variety of severe motor and sensory dysfunctions.

Modelling the mechanics of the mitral valve in the heart is another key application at the centre, performed by our collaborating group in biomechanics at the NTNU in Trondheim. Simulation of a heart valve is a very challenging problem that involves large-deformation, composite material mechanics and fluid-structure interaction. A promising material model for the mitral valve has been published in a series of papers.

We mentioned three publication traditions that the CBC has followed: publication in archival journals, publication of software, and publication of books. Journal papers constitute our primary outlet of results. Nevertheless, our group has a particular focus on the latter two publication channels. One reason for this is that other scientists often find software and books more directly applicable in their research than specialised papers. In this way software and books are effective means for accelerating research elsewhere, and represents an important contribution to science. We believe that over the years the software and books have substantially contributed to our group's increased visibility. In particular, Springer uses some of the books as models for innovative multidisciplinary texts in computational science and engineering, a fact that, together with remarkably good sales, has brought significant attention to Simula's scientific computing activity.

The CBC is also visible at conferences through talks and organising minisymposia. Several of our researchers have been keynote speakers at international conferences, some quite regularly over the years. Scientists associated with the CBC have in total 19 appointments as associate editors in international journals, ten of which are held by full-time employees at Simula. Many also frequently serve on scientific committees for conferences and research councils. Two of our talented young researchers, Joakim Sundnes and Anders Logg, have received the prestigious Outstanding Young Investigator award, which gives them funding over five years, from the Research Council of Norway, to build up their own subgroups. Recently, Aslak Tveito and Hans Petter Langtangen were appointed as editors for the fields of scientific computing and computational partial differential equations in Springer's major new project *Encyclopedia of Applied and Computational Mathematics.*

The last couple of years have also seen the successful recruitment of outstanding young scientists from abroad. Anders Logg from Sweden has been a driving force in several new activities within the CBC, especially regarding the FEniCS software and adaptive finite element methods. Kirsten ten Tusscher from the Netherlands recently established her own subgroup focussing on the mathematical modelling of cell processes and biological evolution.

At the time the Scientific Computing group entered Simula, we were evaluated as excellent in the national evaluation of information and communication technology research. The group received the same evaluation in 2004 and in 2006 the group was awarded the prestigious Centre of Excellence.

What's Next in Scientific Computing at Simula?

Through the sections above we have described a long-term strategy centered around combining work on generally applicable algorithms and software with application oriented research. This balanced focus of the scientific work is combined with a similar balance regarding result outlets, where publication of books and open-source software is combined with traditional journal papers. We believe that this has been a successful formula in the past, and will continue to work along these lines in the future. We will also continue to target challenging applications with a large potential, and in foreseeable future we will continue to choose these applications from the fields of computational biomedicine and computational geoscience. This is in line with Simula's strategy of investigating substantial efforts into large, focussed projects, and staying on these projects for a long time.

As described above, scientific computing is about developing tools and techniques for performing computations in science. On the other hand, computational science and engineering (CS&E) is about applying these tools in specific scientific investigations, as defined in a recent editorial [20]:

> "In fact, as opposed to other mathematical sciences, CS&E often achieves its progress through a clever combination of techniques and methods employed for the different stages of the CS&E pipeline. In such a case, the innovation of excellent research may consist in the creativity needed to synthesize a computational solution for a complex problem from the right building blocks."

Although the application focus has been in existence in the group for more than a decade, the focus on computational science as defined here is of fairly recent date. This is easily seen by looking at the oldest application project in the group, on simulating the electrical activity in the heart. This activity always had a strong application focus, it was substantially different from the generic software activity, and required significant domain-specific expertise from the project team. However, a review of the publications in this field from 2001 to 2006 reveals that they are mostly concerned with development of numerical algorithms and software for solving the involved equations. The project was focused on developing computational building blocks to be used in computational modeling of the electrical activity of the heart, and is most correctly described as a *domain specific* scientific computing project. This focus was a natural one, since the lack of efficient computational methods was (and still is) a major bottleneck in the field, and with the background of the project members this was clearly the area where we could make the most significant contributions. However, we experience that a stronger impact can be obtained by applying the developed methodology to attack the most challenging problems in the field, and thereby demonstrating the performance and usefulness of the tools. This has led to a stronger focus on true computational science, a development that is of fairly new date but is visible in all the application oriented projects. The development results from a strategic decision by the department, and is enabled in part by a stronger inhouse domain expertise, combined with more extensive collaborations with medical doctors and other domain experts.

To summarize, the strategy for scientific computing at Simula remains to keep a balance between generic tools and applications. While past activities in the group have mostly focused on a blend of generic and domain specific scientific computing, current research contains a strong component of true computational science, where the main novelty in the research lies in the computed results, and not in the development of new computational tools. This trend will continue in the future, and we aim to cover the full range from developing generally applicable computational tools to addressing specific, challenging scientific applications. This is obviously an ambitious goal for a research group of the current size, but one which we believe will be feasible through a careful selection of domains and applications, and by setting up effective collaborations with domain experts. We have recently achieved several such effective collaborations, which constitute a promising potential for further development.

The expected benefit from the chosen strategy is threefold. First, the generic tools activity will obviously benefit from close contact with domain experts and real scientific problems, instead of mainly considering simplified model problems for testing and benchmarking. Second, researchers primarily interested in the applications will find the research more efficient and rewarding when building on a set of robust, well tested, state-of-the-art computational tools. Third, the general impact and visibility of the group increases by demonstrating the importance of our computational tools for advancing science. This is also in line with Simula's strategy of performing useful and important research.

The discussion above outlines a clear distinction between scientific computing and computational science, being basically the distinction between computing and science[1], or tools and problems. Our future research strategy of achieving an effective mutual interaction between advanced tools and challenging problems will be of benefit to both subjects: the problems will drive the development of the right tools, and knowing what the tools can do directs the questions that can be answered in science. Over time, we think it is likely that research in computations and tools will be absorbed in the sciences and not exist as a major research field on its own.

References

[1] FEniCS software collection. http://www.fenics.org.

[2] Diffpack software package. http://www.diffpack.com.

[3] H. P. Langtangen. *Python Scripting for Computational Science.* Texts in Computational Science and Engineering. Springer, third edition, 2009.

[4] H. P. Langtangen. *Computational Partial Differential Equations–Numerical Methods and Diffpack Programming.* Texts in Computational Science and Engineering. Springer, 2nd edition, 2003.

[5] J. Sundnes, G. Lines, X. Cai, B. F. Nielsen, K. A. Mardal, and A. Tveito. *Computing the electrical activity in the heart.* Monographs in Computational Science and Engineering. Springer, 2006.

[6] X. Cai, H. P. Langtangen, and H. Moe. On the performance of the Python programming language for serial and parallel scientific computations. *Scientific Programming*, 13(1):31–56, 2005.

[7] H. P. Langtangen and X. Cai. On the efficiency of Python for high-performance computing: A case study involving stencil updates for partial differential equations. *Modeling, Simulation and Optimization of Complex Processes*, pages 337–358. Springer, 2008.

[8] H. P. Langtangen and A. Tveito, editors. *Advanced Topics in Computational Partial Differential Equations - Numerical Methods and Diffpack Programming,*. Lecture Notes in Computational Science and Engineering, vol 33. Springer, 2003. 658 p.

[9] G. T. Lines, M. L. Buist, P. Grøttum, A. J. Pullan, J. Sundnes, and A. Tveito. Mathematical models and numerical methods for the forward problem in car-

[1] The most widespread encyclopedia today, Wikipedia, redirects "Scientific Computing" to "Computational Science" and explicitly demonstrates that these two terms are synonyms. The authors strongly disagree with this interpretation.

diac electrophysiology. *Computing and Visualization in Science*, 5:215–239, 2003.

[10] M. C. MacLachlan, J. Sundnes, and G. Lines. Simulation of st segment changes during subendocardial ischemia using a realistic 3d cardiac geometry. *IEEE Transactions on Biomedical Engineering*, 52:799–807, 2005.

[11] M. C. MacLachlan, B. F. Nielsen, M. Lysaker, and A. Tveito. Computing the size and location of myocardial ischemia using measurements of ST-segment shift. *IEEE Transactions on Biomedical Engineering*, 53:1024–1031, 2006.

[12] J. Sundnes, G. T. Lines, and A. Tveito. Efficient solution of ordinary differential equations modeling electrical activity in cardiac cells. *Mathematical Biosciences*, 172:55–72, 2001.

[13] J. Sundnes, G. T. Lines, K. A. Mardal, and A. Tveito. Multigrid block preconditioning for a coupled system of partial differential equations modeling the electrical activity in the heart. *Computer Methods in Biomechanics and Biomedical Engineering*, 5:397–409, 2002.

[14] M. Hanslien, J. Sundnes, and A. Tveito. An unconditionally stable numerical method for the luo-rudy 1 model used in simulations of defibrillation. *Mathematical Biosciences*, 208:375–392, 2007.

[15] J. Sundnes, G. T. Lines, and A. Tveito. An operator splitting method for solving the bidomain equations coupled to a volume conductor model for the torso. *Mathematical Biosciences*, 194:233–248, 2005.

[16] J. Sundnes, B. F. Nielsen, K. A. Mardal, X. Cai, G. T. Lines, and A. Tveito. On the computational complexity of the bidomain and the monodomain models of electrophysiology. *Annals of Biomedical Engineering*, 34:1088–1097, 2006.

[17] S. O. Linge, G. T. Lines, J. Sundnes, and A. Tveito. On the frequency of automaticity during ischemia in simulations based on stochastic perturbations of the luo-rudy 1 model. *Computers in biology and medicine*, 38:1218–1227, 2008.

[18] A. Tveito and G. T. Lines. A condition for setting off ectopic waves in computational models of excitable cells. *Mathematical Biosciences*, 213(2):141–150, 2008.

[19] A. Tveito and G. T. Lines. A note on a method for determining advantageous properties of an anti-arrhythmic drug based on a mathematical model of cardiac cells. *Mathematical Biosciences*, 217(2):167–173, 2009.

[20] C. Johnson, D. Keyes, and U. Ruede. Special issue on computational science and engineering. *SIAM Journal on Scientific Computing*, 30(6):vii–vii, 2008.

19

CATCHING THE BEAT

An interview with Kirsten ten Tusscher by Dana Mackenzie

One of the most common ways for heart disease to manifest itself in humans is called arrhythmia—an abnormal heartbeat. For instance, the heart may beat too rapidly (tachycardia), or it may flutter very weakly and irregularly (fibrillation). One of the greatest challenges of computational biology is to understand how a physical disease process—such as the formation of scar tissue in the heart—leads to a disruption in the heart's normal timing.

In a normal heartbeat, a wave of electrical activity spreads from the heart's pacemaker node down through the atrium, and then back up from the bottom of the ventricle. The wave front is more or less planar. But in tachycardia, the planar wave folds in on itself, and eventually forms a rotating spiral wave that ignores the heart's normal pacemaking signal. This self-sustaining spiral wave rotates faster than a normal heartbeat, giving the heart less time to fill with blood and pump effectively. In some cases, the spiral may anchor onto a scar in the heart tissue, making the tachycardia more stable. In other cases, a spiral wave may break over scar tissue like an ocean wave breaking over a rock, and a chaotic pattern of electrical activity results—the sign of fibrillation. But fibrillation can also occur without scar tissue, depending on the conditions. The chaotic electrical signals cause local, unsynchronized, contractions of heart muscle, instead of a synchronized contraction of the entire atrium or ventricle. In the case of ventricular fibrillation, this condition leads to sudden death if not treated immediately.

Mathematical models have been remarkably successful at reproducing these phenomena (which have been observed to some extent in animal experiments). In fact, one of the best signs that we really know what is going on in the heart is that we can take a set of mathematical equations that describe what is going at the cellular level (principally the motion of sodium, potassium, and calcium ions into and out of a cardiac cell), then link all of these cells up into an anatomically accurate heart, set the equations going and produce a qualitatively accurate simulation of a guinea pig,

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_19,

or dog, or human heartbeat. We can simulate normal heartbeats, spiral waves, and fibrillation.

But for future clinical applications—for the design of anti-arrhythmia drugs, for instance—qualitative plausibility is not enough. Cardiac models need to be quantitatively accurate, and they aren't there yet. The first heart cell model, by Denis Noble in 1962, was fundamentally flawed because it did not even include the calcium current, which had not been discovered yet. In 1991, Ching-hsing Luo and Yoram Rudy developed a cell model for guinea pig cells that became a standard module for whole-heart models of other animals. For several years, the Luo-Rudy model was the best available cell model because detailed information on human heart cells *in vivo* was hard to come by.

Finally, in 2004, Kirsten ten Tusscher of the University of Utrecht and her doctoral advisor, Alexander Panfilov, published the first human heart cell model that was designed to be simple enough to incorporate into a whole-heart model. The ten Tusscher model has already been widely cited and has even been used in pharmaceutical research, although as you will read below, the modeler herself feels that this application is premature.

In 2008, ten Tusscher accepted an offer to come to Simula Research Laboratories. Her expertise in the forward modeling problem—computing the heart's response to different initial conditions—complements Simula's already existing expertise in the inverse modeling problem. Although her cell models are still too complex to be used for inverse modeling, clearly it is not too unreasonable to hope that the two research programs will interact in the future.

What Simula did not realize, when first contacting ten Tusscher, was that she also had plans to venture out into a field of research where she had not published yet—the study of evolutionary biology. The challenge of updating Darwin's theory for a post-genomic era, as ten Tusscher describes in the interview below, sounds nearly irresistible, and particularly appropriate for Darwin's bicentennial year.

While ten Tusscher will continue her research into cardiac modeling, along with postdoctoral assistant Molly Maleckar, Simula agreed that she could also spend part of her time doing research on evolutionary modeling, with PhD student Tim Dorscheidt. This will give Simula new competence in a field that, on the surface of it, seems far from a real-world application. However, who can say what the future will hold?

"Kirsten, could you talk about your background, and where you got your degree?"

"Contrary to most people at Simula, I'm a biologist. When I came from high school I was interested in a lot of things, and I figured that biology has it all. It has the chemistry, the physics, the math, and of course it has the biology. I was always very much interested in the complexity of biology and how does all of this diversity come about. But after my first year at the University of Utrecht I realized I didn't fit in too well, because most biologists are very non-technical. They like the animals and the plants but they don't like math or physics.

"At Utrecht there is a nice group in theoretical computational biology. I knew this was an obligatory course in the second year. I decided that if I liked it, I would go for that specialization, and if I didn't like it I would switch subjects. I took that course and I very much liked it, and I decided that was the direction for me. I could study biology, but in a much more exact manner than most of my fellow students would do.

"I studied computer science for a year to learn programming and algorithms and more math and physics, then I came back and did a master's specialization. I did one research project there, with Paulien Hogeweg, and one in an informatics department in Amsterdam. I skipped the typical subjects in order to do more bifurcation analysis, nonlinear dynamics, and modeling complex biological systems. I'm predominantly a biologist, but my friends who are biologists think that I am a very strange biologist because I can't determine what any plants are or anything!

"I really like programming and simulating rather than waiting for a long time for an experiment which fails. I had friends who had to grow seeds and wait for the plants to grow, and then there was a fungus on the plants and they had to restart and wait another couple of weeks till the plants had grown. I can just restart the simulation."

"You can grow a plant a lot faster on the computer, can't you?"

"Well, it depends on the application. In the work that I do now, sometimes my simulations are a lot slower than real-life."

"How did you start working on arrhythmias?"

"Going to the cardiac arrhythmias was kind of a coincidence. The group where I come from is now very big, but at that time they were quite small and there was not so much funding. Sasha Panfilov had come to Holland seven to ten years before I became a student. He was from the Puchino Institute, close to Moscow, and I think Zhabotinsky was there as well[1]. They did both physical experiments and modeling with excitable media. He had very diverse interests in biology, complexity and evolution and pattern formation. I decided to apply to work with him. I thought probably he wanted a physicist, but he was very happy for me to apply and we started a nice collaboration."

"When did you start working with Panfilov?"

"I think it was in 1999 or the beginning of 2000 that I started my PhD. Then I worked with him for quite a long time, until I came here. Until I moved here I was quite stuck in one place, doing my PhD and postdoc at Utrecht.

"In cardiac arrhythmia there are a lot of things going on, but what I chose as a niche was human-specific modeling. Sasha had done a lot of whole-heart modeling, but they were on an abstract model or a dog heart. A lot of people were combining

[1] The Belousov-Zhabotinsky reaction was the first chemical reaction to demonstrate complex pattern formation, including spiral waves that look not too dissimilar to the spiral waves that occur in heart tissue.

these models with cell models that were originally devised for guinea pig cells, so you have strange things going on.

"You want to see whether things that apply for animals also apply to humans, because you're interested in saving humans. The first problem I ran into was that there was not an ionic model for human heart cells. So I just decided to do that, and now this model has been cited over 160 times in four or five years. It's living a life of its own. I wanted to put it in a whole heart.

"Even though the numerics that I used back then are not as sophisticated as what they use here, it's still immensely slow, because you have 13.5 million points in this heart. Each of these points has 16 to 20 variables. To be stable, to not blow up numerically, you need to update each one every 0.02 milliseconds. That's why on a computer we're actually slower than life. Even if I compute five seconds of a heartbeat, with twenty computers at the same time parallelizing it, it takes me two days.

"That's why I wanted to stick to it as a postdoc. In some labs, somebody does the cell modeling, someone else does the graphics, and someone else does the parallelization. I do it all on my own. I had invested so much time in the technical stuff that I still wanted to harvest the fruits of my labor in the postdoc phase. Sasha had written this project of putting a whole heart together using this software. I felt that this was my baby, and I didn't want some other postdoc to work on that! At the end of the PhD things start accelerating, the rate at which you publish articles and get results, because all of the supporting software is in place."

"You've gotten the snowball rolling."

"Yes, now we were at the stage where we can do some really detailed quantitative simulations, and we knew some people in England who had clinical data. Then we could really ask the question: 'How does arrhythmia in these human hearts compare with animal hearts?' Of course it's very useful to investigate general things, like how do things become unstable or how do things change if you have ischemia, no oxygen supply. But if you want to make predictions, if you want to make a drug that will enhance this or suppress that, then of course you have to have something more human-like. By now there are some other human models, but they are not specifically tailored for whole-heart simulations. You need a whole supercomputer to do one cell, so it doesn't make much sense to compute a whole heart."

"Can you say what you've found out from your comparisons with clinical data?"

"Sasha investigated how the number of spiral waves depends on the size of the heart. The funny thing is that people thought if you have a larger heart, you have more complex and chaotic behavior. Basically the end result of my model is that the human heart is simpler! Look at this picture of a pig model. You see lots of spirals. Now, here is a polar map of the whole human heart during fibrillation. This is clinical data!

"Believe it or not, people in England were willing to let doctors pull a sock over their heart during open-heart surgery and do measurements, and stay open-chested

Kirsten ten Tusscher

in cardiac surgery a little longer. During the surgery, ventricular fibrillation was induced. I was shocked to hear this, but it seems to be quite normal. If you're installing a defibrillator, you want to make sure that it works. I have been assured this was normal procedure, and the only extra risk was that the patients remained open-chested a little bit longer.

"One of the things you notice in the data is that there are many fewer of these spots[2]. One of the other things is that if we look at the frequency of ventricular fibrillation in humans, it's like 4–5 hertz, but in pigs and dogs it's 10–13 hertz[3]."

"What got you interested in coming to Simula?"

"It worked the other way around. They came to me. They had seen both my model of single-cell stuff and my whole-heart stuff. Per Grøttum, who has a part time position here, apparently discussed with Aslak Tveito that I would be interesting for them and it would be nice for me to give a presentation. They asked for two, one on single cells and one on my whole heart work. And I gave them, and at the end of the day they just offered me a job!"

"Did you accept it right away?"

"No, by that time I was doing other research on evolution, more with Paulien Hogeweg than with Sasha Panfilov. Aslak said to me that if I was interested in working with them, I should send them an email. I said it was definitely an interesting option because my postdoc contract was ending in six months. In Holland, as in most places, to keep going in research you're supposed to go abroad for awhile. So I thought for a while and decided to go for it."

"What were your reasons?"

"If you go abroad, most opportunities are in the States. I didn't want to go that far, because you're so far from family and friends that you're kind of disconnected from

[2] i.e., centres of the spiral waves.

[3] At this point the explanation becomes fairly technical. First, ten Tusscher verified the clinical count of spiral waves with her model. In fact, she demonstrated that the surgeons were undercounting slightly, because some of the waves can only be seen from the inside of the heart looking out. Even so, the undercount was not nearly enough to explain the difference between human and pig hearts. Next, ten Tusscher modified various aspects of the cells' behavior to make them less human-like, and see which changes would most affect the number of spiral waves. She found that the most effective change was in the "action potential duration," which is the minimum length of time that a heart cell can remain excited. In effect, it takes longer to de-excite a human heart cell than a pig cell. Therefore the spiral waves turn slower, the spirals have to be larger, and not as many spirals can fit into one heart. This is good news–it means that the human heart does not fibrillate as readily as it might otherwise. Interestingly, the action potential duration had been hypothesized as an important ingredient in determining whether fibrillation does or does not occur–but the mechanism was believed to be slightly different.

them. It would be nice to see them more than once a year. Also I have a boyfriend who was still getting his PhD, and he was still stuck in Holland.

"So it was convenient to go to a country that was not so far away. Scandinavian countries also have a very nice reputation... The way that people think is quite similar to Holland, it's egalitarian and quite liberal. And Norwegians have a reputation for being very sporting, and I'm a passionate runner. I thought probably that Norway will fit quite nicely who I am and the kinds of things that I am interested in. Indeed, most of the friends I have made in the last six to seven months are from my running club.

"Also, I am getting a lot of freedom. One part of my accepting the job was that I asked if it would be okay to work both on the cardiac research and the evolutionary stuff. And they agreed on that. In science, if you don't have a track record in an area and don't have any publications you can show, the chances of getting money are almost zero. Here, now, I can do both and there's no problem. I can gradually start building publications in that area as well.

"Simula also has great computer resources, of course. I can just put on 30 or 40 simulations without thinking about it, and it's no problem."

"How do they compare to university computers?"

"I guess for some people it might be better here, but for me I was quite spoiled in Utrecht. We had this self-taught system administrator who had built his own Beowulf cluster, and no one was computing on it, so it was just mine. I was sharing the same room with him, so I knew when he was going to shut it down, and I could say, 'Don't do it today.' I was quite spoiled.

"In general it's quite nice here to have your own in-house computer. I've also been at the supercomputer centre in Amsterdam, and that would just be horrible. It's very nice if your simulation speeds up from three weeks to five hours, but then if you have to be in a queue for three weeks it doesn't help much. Or if somebody changes your root path and all of a sudden your simulation can't find a startup file, you've waited three weeks and it crashes immediately! In that sense it's always very nice to have people who run things just across the hall. And it's not very full yet, the cluster here, so it's great."

"As far as your research, what goals do you have?"

"Looking at different mechanisms of arhythmia. Thus far I have looked just at one mechanism. Related to that, I would like to incorporate still more realism in the models. Right now I have a human cell model, but the heart is completely homogeneous. You don't have any cells other than heart cells, whereas we know that there is connective tissue that is needed to give the heart its shape. The amount of connective tissue increases when you have diseases or get older. Usually people only look at the electrical signal to see whether things go wrong, but now we know that intracellular calcium, which leads to contraction of muscle cells, is also important. A lot of people are doing these things, but often in such a way that you cannot use these models in a whole heart. I want to develop models always in such a way that I can do them on the whole heart level. On the one hand I want to have more biological detail, but always to keep in mind that it should be computable. And I want to do it specifically for humans.

"In these models there are so many things to consider. You can look at blood flow. You can look at metabolism, or the influence of the nervous system, or you can look at contraction and the feedback you get from contraction. A fundamental problem is: Do you want to put everything into one model? I have been raised as a theoretical biologist to believe that models should be simplifications, and that you should see if something matters, instead of piling everything into one model. On one hand you can analyze them better if they are simple. On the other hand, if they fail and they fail miserably, you know that you're missing something.

"Some people are now focusing on coupling electrical stuff to mechanical stuff, but as for me, I'm going for the connective tissue and the calcium. If you try to do it all, you get these huge monster models and it's hard to say what's causing what."

"Explain how the connective tissue affects the way the model behaves."

"Put very simply, the cardiac cells are electrically active, so if they generate an electrical signal the next cell senses it, it also produces an electrical signal and so you get this wave going. Connective tissue cells are just different cells, and they make proteins that they deposit outside cells. These proteins are just barriers. They don't generate any electricity, and they don't let through any electricity, so they form blocks for the wave fronts.

"Normally, the connective tissue forms a scaffolding or a matrix for the body. But if you have diseases or get older, heart cells get damaged or die. The problem with differentiated cells like nerve, brain, and heart cells is that they don't divide that readily any more if you're an adult. If a cell dies, its space usually fills up by other cell types proliferating and connective tissue being deposited there. If you look at a very microscopic level, you see small disturbances in your electrical wave. On global level you wouldn't notice that. If you have a disease or you're aging, it increases very much. On top of that you have other problems. Certain ionic currents get more or less, the cardiac cells are not so coupled any more, and you can get local blocks or delays in this electrical wave. It's the electrical wave that tells cells when to contract. So instead of one nice wave front that tells everything when to contract, you get local delays and the whole synchrony gets lost."

"Is there a clinical relevance to that?"

"It's clinically very relevant... It is well known that all kinds of diseases that are associated with increased connective tissue are also associated with increased levels of arrhythmogenesis. There is a lot of clinical research on trying to prohibit cardiac tissue from forming so much connective tissue. It's an inflammatory response to cardiac cells dying, so there's a lot of practical research going on there."

"Do you have particular applications in mind for this research?"

"I'm a basic researcher. That's the short answer. Of course I hope to contribute to insights in arrhythmias that can help make better drugs or treatments. But in the sense that I want to commercialize things, it's not my primary interest. I know that my cell models are used by other researchers to predict about drugs. I'm always very reluctant to see that, because models are never complete. They are always simplifications. I don't think they can yet be used for drug testing."

"Can you tell us about your new research on evolution?"

"Evolution was my first love in biology! I read *The Origin of Species* when I was 16. Darwin was saying that you need heredity, mutation, and competition, and then you get natural selection and gradually fitness goes up in the population.

"There is a relatively new idea in evolution that was not applied in classical population genetics, which is that the mapping from DNA to genome to proteins to cells to organisms is very complex and nonlinear. You can have genomes that are very different and give similar organisms, or very similar and give different organisms. So the genome is not just a 'bag of genes,' as it was seen until recently.

"A question that I'm interested in is speciation. How do you get a species to split up? The first thing I showed is how you can get two populations that are not really different species, just different morphs or different shapes of the same species. For example, you can have two different colors of cichlids in the same lake. This doesn't make sense if you assume you're just a bag of genes. In a classical population genetics model, if you have a blue parent and a red parent, the child will always be purple. Unless the formation of morphs and species is simultaneous, you can't actually get to a population of red and a population of blue, because you will keep collapsing back to purple.

"The essential part of my model is that even without separating into two species, you can still have this differentiation into different morphs. Because of the need to blend into the background, it may be advantageous for a fish to be red or blue, while a purple fish may have very low fitness. By using the genome and the gene regulatory network in a clever manner, you can largely resolve that problem. The child of the red and blue parent can become very reddish or very bluish. They can be quite good at what the parent is doing, although never 100 per cent. This is a simple way of having different forms in one population.

"In the current model setting, if I give the animals the ability to develop a preference to mate with others of the same type, they will; it will always be more advantageous to become real biological species rather than two morphs within the same species. But in my model the two things do not have to occur at the same time. In the classical model, it has to be simultaneous."

"What does a gene regulatory network look like? How does it differ from the 'bag of genes' model?"

"You can think of a graph with a lot of nodes, representing the genes, and links that represent the interaction of genes. This gene is repressing that one, but that one is activating the first one, and so on. It's like a jungle of genes and interactions.

"For each node you can have an activity level for that node, so you can model that with a variable. Then you get a very complex set of coupled ordinary differential equations with many nonlinear terms, because it is this protein binding to a site on the DNA that affects that protein, so you get a lot of Michaelis-Menten saturation constants.

"In classical population genetics they used differential equations, too, but the equations are often linear. This means that if you want to have very different looks for an organism, you will also need a different genome. We know now that is not true. In the gene networks, with all these interactions, first of all you have lots of differential equations, so you cannot solve them analytically–you have to do it numerically. Also, because they are not linear, a small variation on the genome can be amplified, through nonlinearity, to have a very unexpected effect. All of a sudden your animal has a horn, or teeth where it didn't before."

"Are there other people doing these kinds of models?"

"Gene network modeling has not previously been done for speciation research, as far as I know. Gene network models have been used quite a lot to model differentiation and pattern formation in multicellular organisms. (Different cells form different colors, even though their DNA is the same, because the regulatory networks switch on different genes in the same animal). In the literature on pattern formation, people have mostly looked at only two types of cells (for example, black and white hair cells forming a pattern of stripes) and hence two different combinations of genes turned on and off. I have been looking at the evolution of a range of different cell types in a row from head to toe, forming the body plan of an organism."

"Darwin would be very happy, I think, to see that he didn't actually finish evolution!"

"People say that there is a new synthesis needed. After Darwin, when they rediscovered Mendel, there was the synthesis of Darwin's theory with population genetics. Now you see that simple population genetics is not enough, because you have to consider this whole mapping between the gene network and the phenotype, which involves cell differentiation and development, and also the impact of the environment on the characteristics of the organism. That is why people are calling for a new evolutionary synthesis, so that evolutionary theory not only involves Darwin's original theory and population genetics but also gene regulation, developmental biology and ecology. Evolutionary biology is far from finished, and lots of interesting changes are currently occurring!"

20

COMPUTER SIMULATIONS OF THE HEART

Glenn T. Lines and Joakim Sundnes

Glenn T. Lines · Joakim Sundnes
CBC, Simula Research Laboratory

Glenn T. Lines · Joakim Sundnes
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_20,

PROJECT OVERVIEW

Cardiac Computations

The heart is our most important muscle. It provides the blood circulation necessary to sustain life. It pumps 7000 litres per day and in an average lifespan it pumps the equivalent of the payload of a super tanker. Diseases affecting the heart's performance can be very serious, since the body vitally depends upon proper circulation, and heart disease is currently the world's leading cause of death. Understanding more about how the heart works will improve the quality of diagnosis and treatment, and computer simulations have emerged as a powerful tool to increase this understanding. We envision that the significance of computer models will continue to grow in the future, and that they will be an important supplement to traditional experimental techniques.

Scientific Challenges

The propagation of the electrical signal in the heart can be described by a system of partial differential equations known as the bidomain model. This system is coupled to systems of ordinary differential equations that describe electro-chemical reactions in the heart cells. The total system of equations is known to be challenging to solve on a computer, primarily because of the complexity of the involved equations and the rapid dynamics of the electrical activation process. Computing accurate solutions on a human sized heart is a huge computational task, both in terms of memory and CPU usage. In spite of vast improvements in both numerical methods and computer hardware, computational load is still a severe bottleneck for progress in this field.

Extending the mathematical model with a description of the mechanics of the muscle further adds to the complexity of the system. Although the resolution requirements for the mechanics part are known to be much less strict than for the electrophysiology, this part introduces additional challenges such as large deformations and strongly non-linear and anisotropic material behavior.

Obtained and expected results

Computational modeling of the heart has been a research topic at Simula since the start in 2001. The main focus has been on the electrical activity, and in particular on deriving efficient numerical methods for the bidomain model. The Simula group was the first to demonstrate, through analysis and numerical experiments, order optimal behavior of multilevel solvers for the bidomain model. Since the discretization of the bidomain model typically leads to huge linear systems to be solved, this is a particuarly important result. We have also been influential for the introduction of operator splitting methods in the field, methods which are now regarded as the standard methods for this problem. The work on operator splitting has also been extended to include the fully coupled electro-mechanics problem.

More recently, the focus of the heart modeling research at Simula has changed slightly. Instead of focusing entirely on development of new numerical methods, the research is now moving in the direction of applying the methods to study specific biomedical problems. Recent results may increase our understanding of cardiac arrhythmias in general, and may also be directly applicable in the development of anti-arrhythmic drugs.

COMPUTER SIMULATIONS OF THE HEART

Biomedical research has traditionally used two types of experimental techniques–*in vivo* experiments performed on living organisms and *in vitro* studies performed on tissue samples. A third technique called *in silico* experiments is emerging. As the name suggests, these experiments are performed on a computer, that is, in silicon. In this paper we will present a project that studies the heart using in silico experiments.

Roughly speaking, an in silico experiment is performed by formulating a mathematical model of the phenomena one wants to study, creating a piece of software that solves the governing equations, and finally running the code on an appropriate computer and then interpreting the results. This is typically an iterative process where the outcome from a simulation might generate new questions that call for modification of the mathematical model.

One obvious benefit of in silico studies is that they are normally cheaper to perform than physical experiments. In some cases it may even be impossible to undertake the corresponding physical experiments due to practical, economical, or ethical reasons.

In order to draw certain conclusions from the simulations, the underlying mathematical model must be thoroughly validated. Simulations are useful, however, even when the underlying model is validated only to a limited degree. For example, one might formulate a set of equations governing the interactions of different proteins. Through simulation of this model one can obtain testable predictions that can be compared with observations. A mismatch would indicate that something is wrong with the understanding of the system. In this way in silico experiments can complement traditional experiments.

20.1 Mathematical Model

Our aim is to study the *electrical* activity of the heart, so the primary unknown is the electrical field generated by the heart during a cardiac cycle. The mathematical model we are going to use is based on Maxwell's electromagnetic laws. It states that the electrical field E is related to the magnetic field B in the following way:

$$\nabla \times E + \frac{\partial B}{\partial t} = 0.$$

For the application under study this model is unnecessarily detailed. Specifically, the magnetic fields vary so slowly in the body during a heart beat that setting $\dot{B} = 0$ is a reasonable approximation. This is called the quasi-static assumption, *quasi* since the field is not really constant in time. With this reduction we end up with the following simple law for the electrical field:

$$\nabla \times E = 0.$$

This relation implies that the field can be written in terms of a potential function u like this:

$$E = \nabla u.$$

Furthermore, if we assume that the material we study behaves like an ohmic resistor, the current density J can be written in terms of E in the following way:

$$J = ME,$$

where M is the conductivity of the medium. In general, M is a tensor which, for the case of muscle tissue, will depend upon the orientation of the muscle fibres. Conductivity is highest along the fibre directions and lowest across the fibre direction.

Assuming that current is conserved inside the domain, the divergence of the field is zero:

$$\nabla \cdot J = 0.$$

Inserting the above expression for the current, we end up with an elliptic equation for the electrical potential:

$$\nabla \cdot M \nabla u = 0. \tag{20.1}$$

In the torso, outside the heart, it is acceptable to assume that the tissue behaves like an ohmic resistor and that there are no current sources. Hence, equation (20.1) is a good model for the electrical potential in the torso. These assumptions are not, however, appropriate for the heart. Instead, we use a model known as the bidomain model. This model was originally proposed by Tung[3] and is regarded as an accurate representation of the electrical properties of the myocardium. In the model the tissue is divided into two domains, the intracellular space and the extracellular space, and each domain is assumed to behave like an ohmic resistor. Current is not conserved inside each domain, because it is allowed to flow between the two domains. This current must cross the cell membrane and is therefore called the transmembrane current. The total current of the two domains is still conserved.

These assumptions imply the following model:

$$\nabla \cdot J_i = -\nabla \cdot J_e,$$
$$\nabla \cdot J_i = I_m,$$

where J_i is the current in the intracellular space, J_e is the current in the extracellular space, and I_m is the transmembrane current. The first equation states that the inflow in one domain is equal to the outflow of the other domain. The second equation states that the outflux is equal to the transmembrane current.

The transmembrane current has a capacitive and a resistive part, the latter representing ion flow across the membrane:

$$I_m = C_m \frac{\partial v}{\partial t} + I_{ion},$$

where v is the transmembrane potential, defined as the difference between the intra- and extracellular potentials. Denoting the intracellular and extracellular potentials by u_i and u_e, respectively, we have $v = u_i - u_e$. We assume ohmic behaviour of the currents in the two domains. Substituting for J_i and finally eliminating u_i from the equations, we arrive at the following form of the bidomain model:

$$\nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e) = C_m \frac{\partial v}{\partial t} + I_{ion}, \tag{20.2}$$

$$\nabla \cdot (M_i \nabla v) + \nabla \cdot ((M_i + M_e) \nabla u_e) = 0, \tag{20.3}$$

where M_i and M_e are the conductivities in the intra- and extracellular spaces, respectively.

The ionic flow I_{ion} depends not only on the transmembrane potential but also on the conductance of the membrane and the concentration of ions on either side of the membrane. If we collect these entities in a vector and denote it s, we have $I_{ion} = I_{ion}(v, s)$. The variables in s are typically modelled with ordinary differential equations (ODEs), that is,

$$\frac{\partial s}{\partial t} = F(s, v). \tag{20.4}$$

There is a great variety in these cell models. Some are aimed at just reproducing the shape of the action potential while the more realistic ones describe many other aspects of the cells, such as the states of channel proteins and calcium buffering. For the simulations in this paper we used a model by Winslow et al. [4], which has 31 state variables.

Model summary. The model we use for the electrical activity consists of the bidomain model (20.2–20.3) in the heart, the elliptic equation (20.1) in the torso, and an ODE model (20.4) to compute the ionic flow. In order to have a complete model we also need boundary conditions. On the heart-torso surface the extracellular potential and current are assumed to be continuous, whereas a no-flow condition is enforced on the intracellular current. On the torso surface a no-flow condition is used, since air acts as an insulator. Summing up, we have the following model:

$$\begin{aligned}
\frac{\partial s}{\partial t} &= F(s,v) && x \in H && (20.5)\\
\nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e) &= \chi \frac{\partial v}{\partial t} + I_{ion}(s,v) && x \in H, && (20.6)\\
\nabla \cdot (M_i \nabla v) + \nabla \cdot ((M_i + M_e) \nabla u_e) &= 0 && x \in H, && (20.7)\\
\nabla \cdot (M_T \nabla u_T) &= 0 && x \in T, && (20.8)\\
u_e &= u_T && x \in \partial H, && (20.9)\\
n \cdot (M_i \nabla v + (M_i + M_e) \nabla u_e) &= n \cdot (M_T \nabla u_T) && x \in \partial H, && (20.10)\\
n \cdot (M_i \nabla v + M_i \nabla u_e) &= 0 && x \in \partial H, && (20.11)\\
n \cdot M_T \nabla u_T &= 0 && x \in \partial T, && (20.12)
\end{aligned}$$

where the heart domain is denoted H and the torso volume outside the heart is denoted by T. The subscripts i and e refer to the intracellular and extracellular space, respectively, and the subscript T refers to entities defined in the torso domain.

20.2 Numerical Methods

Several challenges arise when performing numerical simulations based on the mathematical model described. An obvious challenge is the complexity of the model, which consists of a system of nonlinear partial differential equations (PDEs) coupled to a large system of nonlinear ODEs. Other challenges include irregular geometries and complicated material properties.

Electrical activation of the heart cells is a very fast process, where the polarity of the cell membrane completely changes in just a couple of milliseconds. On the tissue level, signal propagation takes the form of a wavefront of depolarisation propagating through the muscle. Because of the fast dynamics of the depolarisation process, this wavefront is very narrow, typically around 1 mm. These large variations in space and time require numerical solutions of the model to use a very high spatial and temporal resolution.

Through numerical experiments (see, e.g., [5]), we have found that in order to obtain a suitable accuracy, a time step length of approximately 0.1 ms and a nodal distance of about 0.2 mm are required. For simulations on a full-scale human heart, the nodal distance requirement leads to approximately 40 million nodes. Realistic models for cellular activity may contain up to 50 state variables, resulting in about $2 \cdot 10^9$ unknowns that must be determined for each time step. A complete heart cycle is approximately one second and, with the given temporal resolution, this amounts to about 10,000 time steps. Simulation of a complete heart cycle therefore involves approximately $2 \cdot 10^{13}$ degrees of freedom.

Because of the massive computational work required for full-scale simulations of even a single heart beat, this kind of simulation is not yet practical to employ in biomedical research. With the rapid development of computer hardware and numerical algorithms, however, the task is about to become within reach, see, e.g., [6] for an example of large-scale parallel bidomain simulations. However, although accurate full-scale simulations are within reach, the computation times need to be substantially reduced before they are widely applicable research tools. As described for instance in [7], this can only be achieved by combining parallel hardware with the most efficient numerical methods.

Handling Complexity. As stated previously, the bidomain model coupled to realistic cellular models is a highly complex mathematical model, for which it is difficult to develop efficient solution techniques. One approach for reducing its complexity is to employ some form of operator splitting, methods that split a complicated system into smaller parts for which it can be easier to derive efficient solution algorithms.

There exist a large number of different operator splitting techniques and many of them are suitable for the bidomain model. We have chosen to focus on a group of methods known as fractional step methods (see, e.g., [2]). With these techniques, the bidomain model is split into the two subproblems:

$$\frac{\partial s}{\partial t} = F(s,v) \qquad x \in H, \tag{20.13}$$

$$\frac{\partial s}{\partial t} = -I_{ion}(s,v) \quad x \in H, \tag{20.14}$$

and

$$\nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e) = \chi \frac{\partial v}{\partial t} \quad x \in H, \tag{20.15}$$

$$\nabla \cdot (M_i \nabla v) + \nabla \cdot ((M_i + M_e) \nabla u_e) = 0 \qquad x \in H, \tag{20.16}$$

$$\nabla \cdot (M_T \nabla u_T) = 0 \qquad x \in T. \tag{20.17}$$

We see that the coupled nonlinear system of ODEs and PDEs is reduced to a system of nonlinear ODEs and a system of linear PDEs. Solutions of these subsystems can be combined using Strang or Godunov splitting techniques (see, e.g., [2]) to obtain an approximate solution of the full bidomain model. The Strang splitting technique enables an second-order accurate solution of the equations.

Although the two problems (20.13) and (20.14) and (20.15)–(20.17) are simpler than the full bidomain model, it is not possible to derive analytical solutions of these systems. We therefore have to employ numerical solution techniques both for the ODE system and the PDEs. To achieve second-order accuracy of the Strang splitting, the solution techniques for the two subproblems must be at least second-order accurate.

Solving systems of ODEs. The challenge of solving the ODE system (20.13) and (20.14) is highly dependent on the level of realism of the involved cell model. For simple models of the FitzHugh-Nagumo type (see, e.g., [8]), the state vector s in (20.13) has only one component. The effort of solving these equations is then reduced to solving a coupled system of two ODEs. Although the ODEs are nonlinear, this system is not very challenging to solve and can be handled efficiently by simple methods such as explicit Runge-Kutta methods.

For more realistic cell models, such as the one described in [4], the solution of the system (20.13) and (20.14) becomes much more challenging. An obvious difference from the simpler models is that s is now a state vector which can have 30 to 50 components. Realistic models also tend to include highly complicated nonlinear rate functions.

An additional challenge of realistic models is that they model cellular processes occurring over a wide range of time scales. This makes the ODE systems stiff (see, e.g., [9]) and therefore challenging to solve numerically. The practical consequence of stiffness is that we experience severe stability problems when applying explicit numerical methods. In fact, for the accuracy requirements relevant to our application, the time step allowed by the ODE solver is completely dictated by stability requirements rather than accuracy. These issues motivate the use of more stable implicit methods.

There exist a large number of standard solvers for stiff systems of ODEs. Commonly used methods include multistep methods such as the backward differentiation formula (BDF) and various implicit Runge-Kutta methods (see, e.g., [9]). The spatial discretization of the bidomain model, which is described in more detail below, typically requires solving one ODE system for each node in the computational grid. For realistic simulations this implies solving millions of ODE systems, which are all of moderate size, see, e.g. [10]. For this task we have found that one-step methods such as Runge-Kutta methods are the most practical to use, mostly for reasons of memory consumption.

Based on the accuracy requirement for our application and the time step dictated by the operator-splitting technique, we find implicit methods of fairly low accuracy to be suitable. Based on numerical experiments, we have chosen to implement a four-stage method of order three, with an embedded second-order method for error estimation (see [10] for details).

Solving the linear PDE system. In addition to the ODE systems, for each time step we must solve the linear PDE system (20.15)–(20.17). As for the ODEs, these equations must be solved to at least second-order accuracy if we are to achieve the second-order accuracy offered by the Strang splitting technique. We have therefore chosen to base the time discretisation of (20.15)–(20.17) on the Crank-Nicolson scheme. This gives a time-discrete PDE system of the form

$$C_m\chi\frac{v^{n+1}-v^n}{\Delta t} = \frac{1}{2}\nabla\cdot(M_i\nabla v^{n+1}) + \frac{1}{2}\nabla\cdot(M_i\nabla v^n) + \nabla\cdot(M_i\nabla u_e^{n+1/2}) \quad x\in H, \tag{20.18}$$

$$\frac{1}{2}\nabla\cdot(M_i\nabla v^{n+1}) + \nabla\cdot((M_i+M_e)\nabla u_e^{n+1/2}) = -\frac{1}{2}\nabla\cdot(M_i\nabla v^n) \quad x\in H, \tag{20.19}$$

$$\nabla\cdot(M_o\nabla u_o^{n+1/2}) = 0, \quad x\in T, \tag{20.20}$$

which must be solved with appropriate boundary conditions. The notation indicates that v is approximated at the end of each time step, while u_e and u_T are approximated at the midpoint of each step.

Because of the irregular geometry of the heart muscle, it is convenient to apply the finite element method for the spatial discretisation of (20.18)–(20.20). The finite element method is based on the weak form of the equations, which is obtained by multiplying the equations with suitable test functions and integrating over the respective domains. A detailed description of the finite element discretisation is given in [5, 11] and we only give a brief summary here.

We introduce function spaces $V(H)$ and $V(T)$, and multiply (20.18) and (20.19) with a test function $\psi \in V(H)$ and (20.20) with a function $\eta \in V(T)$. Integrating over the two domains and applying Green's lemma (see, e.g., [12]), we obtain a weak formulation of (20.18)–(20.20). By using the continuity condition (20.9) on the interface between the two domains H and T, however, this system can be simplified further. If we introduce a function space $V(H \cup T)$, the weak forms of (20.19) and (20.20) can be combined into a single equation. To simplify the notation, we introduce the inner products

$$
\begin{aligned}
(v,\phi) &= \int_H v\phi\, dx, & a_i(v,\phi) &= \int_H M_i \nabla v \cdot \nabla \phi\, dx,\\
a_{i+e}(v,\phi) &= \int_H (M_i + M_e)\nabla v \cdot \nabla \phi\, dx, & a_T(v,\phi) &= \int_T M_T \nabla v \cdot \nabla \phi\, dx.
\end{aligned}
$$

The complete weak formulation can now be written as follows.
Find $v^{n+1} \in V(H)$ and $u^{n+1/2} \in V(H \cup T)$ such that

$$
\begin{aligned}
(v^{n+1},\psi) + \frac{1}{2}\Delta t a_I(v^{n+1},\psi) + \Delta t a_I(u^{n+1/2},\psi)&\\
= (v^n,\psi) - \frac{1}{2}\Delta t a_I(v^n,\psi) \quad \text{for all } \psi \in V(H),&
\end{aligned}
\tag{20.21}
$$

$$
\begin{aligned}
\frac{1}{2}\Delta t a_I(v^{n+1},\varphi) + \Delta t a_{I+E}(u^{n+1/2},\varphi) + \Delta t a_T(u^{n+1/2},\varphi)&\\
= -\frac{1}{2} a_I(v^n,\varphi) \quad \text{for all } \varphi \in V(H \cup T).&
\end{aligned}
\tag{20.22}
$$

We now introduce finite element approximations for v^{n+1} and $u^{n+1/2}$,

$$
\begin{aligned}
v^{n+1} &\approx v_h = \sum_{j=1}^{n} v_j \phi_j,\\
u^{n+1/2} &\approx u_h = \sum_{j=1}^{m} u_j \phi_j,
\end{aligned}
$$

where n and m are the number of basis functions in the spaces $V_h(H)$ and $V_h(H \cup T)$, respectively. Inserting these expressions into the weak formulation above yields a block-structured linear system of the form

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}. \tag{20.23}$$

The blocks are defined by

$$\begin{aligned}
A_{ij} &= (\phi_j, \phi_i) + \frac{1}{2}\Delta t a_I(\phi_j, \phi_i), && i,j = 1,\dots n \\
B_{ij} &= \Delta t \beta_j a_I(\phi_j, \phi_i), && i = 1,\dots n, j = 1,\dots,m \\
C_{ij} &= 2\Delta t a_{I+E}(\phi_j, \phi_i) + 2\Delta t a_T(u^{n+1/2}, \phi), && i,j = 1,\dots m, \\
a_i &= (v^n, \phi_i) - \frac{1}{2}\Delta t a_I(v^n, \phi_i), && i = 1,\dots,n, \\
b_i &= -\Delta t a_I(v^n, \phi_i), && i = 1,\dots,m.
\end{aligned}$$

Note that we have multiplied (20.22) by two in order for (20.23) to be symmetric.

As noted previously, the resolution requirements when solving the bidomain equations are very strict. For simulations on realistic geometries, this leads to a large number of nodes, which in turn implies that the linear system (20.23) will be very large. To be able to perform accurate simulations in a reasonable time, it is therefore crucial that this system be solved efficiently.

We use the preconditioned conjugate gradient (PCG) method to solve the linear system. A block diagonal matrix of the form

$$\mathcal{B} = \begin{bmatrix} \tilde{A}^{-1} & 0 \\ 0 & \tilde{C}^{-1} \end{bmatrix}$$

is used to precondition the system, where $\tilde{A}^{-1}$ and $\tilde{C}^{-1}$ are approximate inverses computed with a multigrid algorithm (see, e.g., [11] for details).

It can be shown that this preconditioner is optimal for the problem at hand. More precisely, if we introduce the notation

$$\mathcal{A} = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix},$$

the condition number of the preconditioned system $\mathcal{B}\mathcal{A}$ is bounded independently of the number of unknowns. Since the number of iterations used by the PCG method is proportional to the square root of the condition number, the number of iterations will also be bounded independently of the number of unknowns. The work in one PCG iteration is proportional to the number of unknowns in the system and the work of solving the linear system therefore grows linearly as the number of unknowns increases.

The optimal preconditioner for the linear system is an important result for the feasibility of large-scale simulations. For a non-optimal solution method, the workload will grow like N^α, with $\alpha > 1$ and where N is the number of unknowns. As noted previously, for realistic simulations N will be very large and the difference between optimal and non-optimal methods then becomes substantial. With the discretisation techniques used here, the workload of solving the ODEs grows linearly with the number of unknowns and optimal preconditioning ensures that the CPU efforts of the two subproblems stays balanced for all values of N.

20.3 Simulations

In the first simulation of this section we will look at the effect ischemia has on the ECG. The computations are performed on a two-dimensional slice of the body (see figure 20.1a). The electrical propagation is triggered at the three indicated locations on the heart surface. The fibre directions of the slice are shown in figure 20.1b.

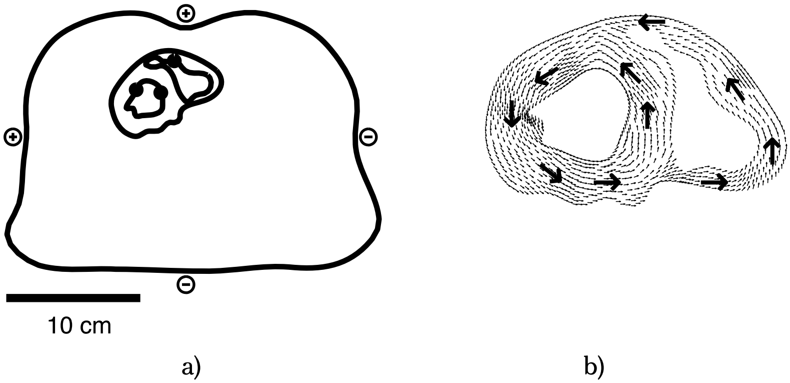


Figure 20.1 a) The heart and torso boundaries. The black dots on the endocardial surface indicate stimulation points. The plus and minus poles on the torso surface show the locations where the surface potential is recorded. b) The fibre directions in the heart. The large arrows are the selected directions and the small arrows are interpolations.

Normal propagation

In the resting state, that is, between heart beats, the tissue has a negative resting potential, typically $v = -90$ mV. As noted previously, the signal propagation in the heart takes the form of a depolarisation, where the potential rises rapidly and reaches a positive peak potential after a couple of milliseconds. A plateau phase follows where the potential remains positive for a little more than 100 ms. Finally the cell repolarises, returning to its negative resting potential. The whole process is called the action potential and typically lasts around 300 ms. Figure 20.3b shows the action potential for a normal and an ischemic cell.

The top row of figure 20.2 shows two time instances during the depolarisation phase, that is, when the cardiac tissue is activated. The bottom row of figure 20.2 shows two snapshots of the repolarisation process. The gradients in the solution are smaller in this phase of the cycle, since the cells activate much faster than they deactivate.

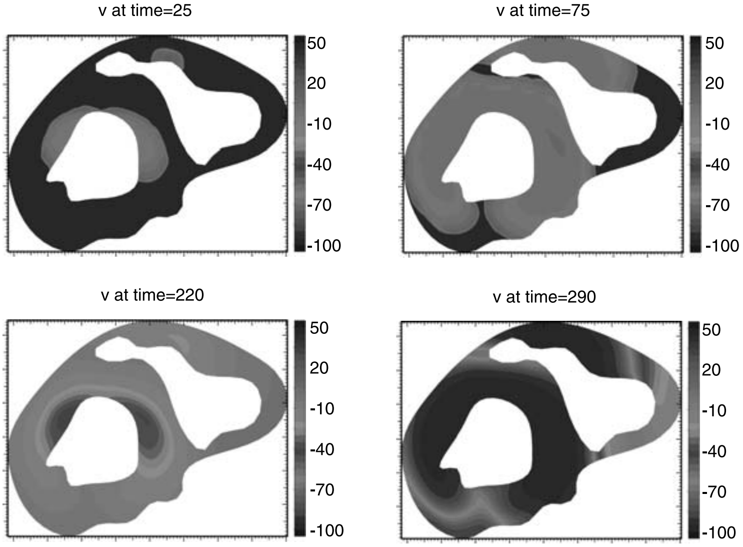


Figure 20.2 The transmembrane potential (mV) at four stages. during normal propagation. (For the colour version, see figure C.3 on page 644.)

Ischemia

We now look at the propagation when part of the muscle tissue is affected by ischemia, that is, some of the cells suffer from a reduction in the blood supply. The affected area is indicated in figure 20.3a. The cellular model in the ischemic area is altered to represents the pathological condition. figure 20.3b shows a comparison of the action potentials in the healthy and the ischemic case. Two notable differences are that in the ischemic case the resting potential is elevated and the action potential has a shorter duration. Below we will see how this influences the extracellular potential and thereby the ECG.

Figure 20.4 shows results with the ischemic area introduced. The first noticeable difference is that the potential during early depolarisation is higher in the ischemic area (see figure 20.4a). After 125 ms the whole heart is depolarised. In the healthy case the heart would be isoelectric at that time, but in the ischemic case we see that the affected area has already started to repolarise. This is due to the comparatively short action potential duration of the ischemic cells, as we saw in figure 20.3b. The effect of different action potential durations is even more pronounced after 200 ms (see figure 20.4c). After 400 ms both the healthy and ischemic tissue have repolarised. Again in contrast to the normal situation, the tissue is not isoelectric (see figure 20.4d). The reason this time is that the resting potential is higher in the ischemic cells.

Let us revisit the states of the simulations at 125 and 400 ms but, instead of considering the transmembrane potential, we now look at the extracellular potential. Figure 20.5 shows a close-up around the ischemic area. The gradients in the transmembrane potential also generate gradients in the extracellular potential. Recall that

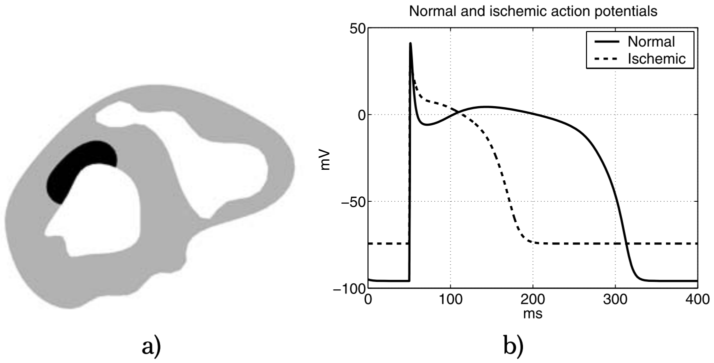


Figure 20.3 a) In the dark area the ODE model is modified to represent ischemic tissue. b) The normal and modified action potential of the Winslow model.

in the normal case there are no gradients in the transmembrane potential at the time instances depicted in figure 20.5 and hence there is no signal in the extracellular potential.

The gradients in the extracellular potential give rise to a current that can be detected on the body surface. The depolarisation and repolarisation phases of the heart show up as strong signals in the ECG and are called the QRS complex and the T wave, respectively. In the in-between plateau phase, called the ST segment, there is normally no signal, since the heart is isoelectric. The gradients in figure 20.5, however, generate the characteristic ST change in the ECG which is associated with ischemia. Figure 20.6 shows the simulated ECGs corresponding to the electrode positions shown in figure 20.1. In the simulation with ischemic tissue (dotted line) we see a clear ST change in at least one pair of electrodes. Chapter 22 presents a simplified version of the bidomain model, which is constructed to model only the ST segment shift during ischemic heart disease. This model is then applied to compute the ischemic region based on a given electrocardiogram.

Simulation on a three-dimensional geometry

As noted in section 20.2, simulation in three dimensions on a full body is a huge computational challenge. Using a single processor computer is currently not an option; there is not enough memory and, even if there were, the CPU time would be prohibitively long. Instead, parallel computers are employed. The results presented here are performed on parallel computers with 512 processors and 512 Gb of memory. The algorithm uses a preconditioner based on domain decomposition for the block linear system arising from finite element discretisations (see [1] for further details).

The heart geometry is based on data from the Bioengineering Institute at the University of Auckland[13], while the torso is generated from slices taken from the Visible Human data [14].

Two snapshots of the solution are shown in figure 20.7.

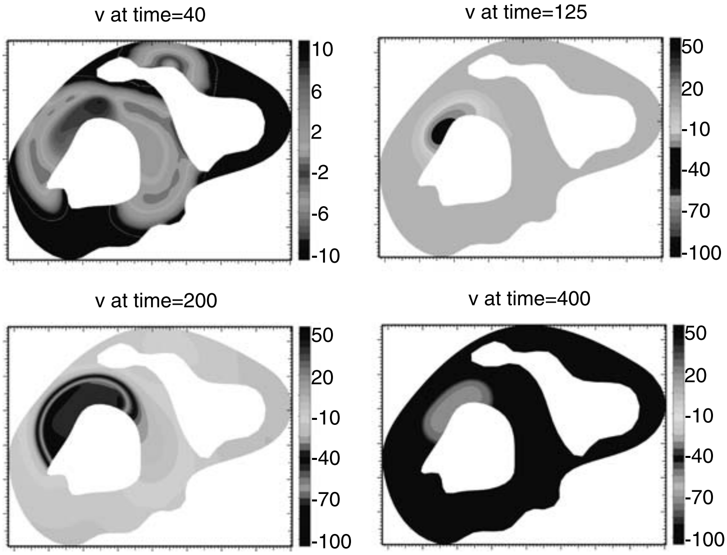


Figure 20.4 Four stages during ischemic propagation. (For the colour version, see figure C.4 on page 645.)

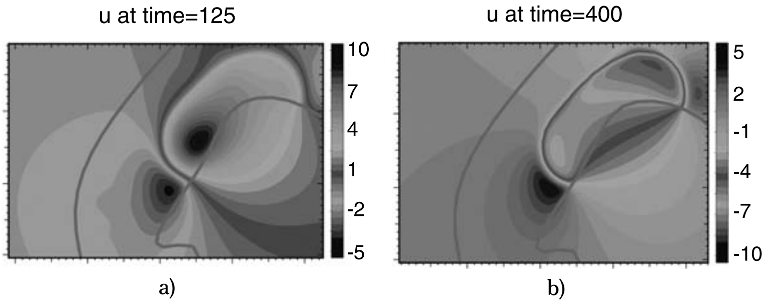


Figure 20.5 The extracellular potential around the ischemic area during a) the ST segment and b) the TP segment. The heart boundary is indicated by the solid line. (For the colour version, see figure C.5 on page 645.)

Table 20.1 shows the CPU time spent on solving the linear systems. The largest system has more than 81 million unknowns. The CPU time increases at about the same rate as the number of unknowns, indicating that the preconditioning is optimal. The speed-up is not perfect, that is, the CPU time is not reduced by 50 per cent when the number of processors is doubled. The reason is that there is some overlap between the domains, which generates an overhead. This problem is most pronounced for the smallest problem sizes.

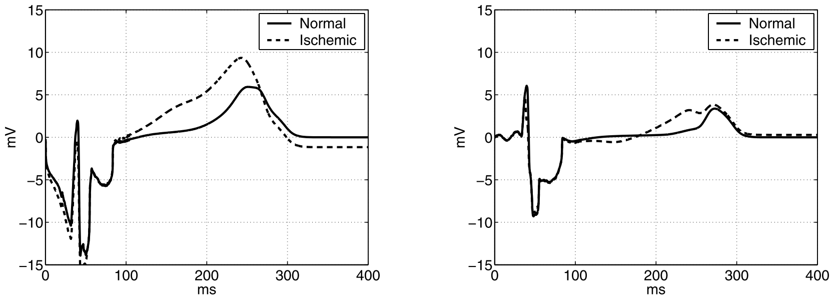


Figure 20.6 The surface potential recorded at the two pairs indicated in figure 20.1. The left panel shows the front-to-back electrodes and the right panel shows the left-to-right electrodes.

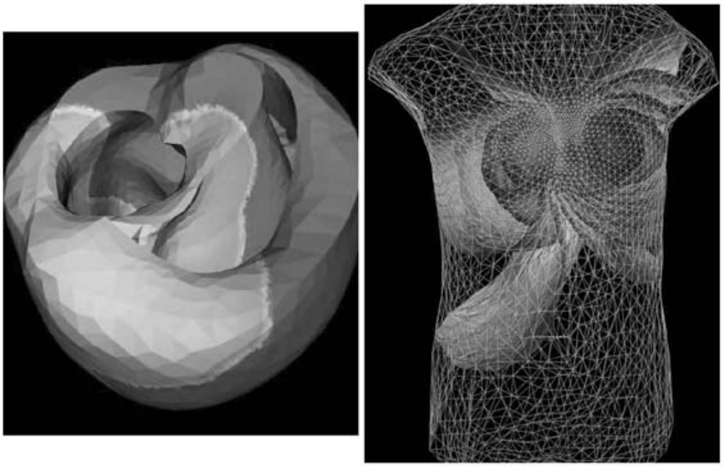


Figure 20.7 A snapshot of the transmembrane potential during the early phase of depolarisation. Red tissue is in the resting phase while the blue tissue is depolarised. The right figure shows iso-surfaces of the electrical potential in the torso. (For the colour version, see figure C.6 on page 646.)

Grid	# unknowns	Number of processors		
levels	$(\mathbf{v}+\mathbf{u})$	$P = 16$	$P = 32$	$P = 64$
3	1,552,283	32.66	46.08	44.11
4	10,705,353	399.64	224.15	148.28
5	81,151,611	4094.97	2915.88	1902.91

Table 20.1 Wall-clock time measurements (in seconds) for solving the linear system(s) for one time step.

20.4 Conclusion and Perspectives

Performing simulations of the electrical activity of the heart poses a huge computational task that calls for sophisticated numerical software. Through the use of modern algorithms such as multigrid and domain decomposition, it is now possible to perform full-scale simulations. Parallel hardware is, however, a necessity and current solution strategies still leaves room for improvement. An approach that has not been discussed here, but which holds great potential for CPU time reduction, is to employ adaptive discretization techniques in space and time. This is particularly attractive for the problem at hand because the solution is very smooth in large areas of the domain, but with large local variatons. A review of recent advances in numerical techniques for the bidomain model, including adaptive techniques, is found in [7].

For many applications, one is not only interested in the electrical activity of the heart but its overall function, including the movement of the muscle and flow of blood inside the heart cavities. With such a detailed model, a wider range of applications can be addressed. The relevant mathematical model will in this case be based on the laws of continuum mechanics and electromagnetism, combined with constitutive relations derived from biomedical experiments. Although obviously useful for investigating a number of clinically important conditions, heart failure being perhaps the most prominent, coupled electro-mechanical modeling of the heart is still in its infancy. There are several reasons for this, including the extreme computational load of the electrophysiology problem, as discussed previously, as well as the complexity of the soft-tissue mechanics models that govern the movement of the muscle. Although the computational load of this part is typically much smaller than for the electrophysiology, the mechanics models introduce additional difficulties in the form of large deformations, complicated coupling of electrophysiology and mechanics, and strongly non-linear, anisotropic material behavior. Some of these difficulties are discussed in [16, 15].

Although in silico models of the heart are extesively used for studying fundamental mechanisms and relations, it is safe to assume that this field is still in its infancy. Through improvement in mathematical models, imaging and data acquisition techniques, numerical methods, computer hardware and software reliability, the predictive power and accuracy of the models is likely to improve dramatically over the next decade. This will pave the wave for in silico models as an indispensable tool in biomedical research, of comparable importance to classical in vitro and in vivo experiments. Thorough validation of the models may also open up for extensive clinical use, where patient specific models serve as test beds for evaluating different interventions and treatments. The topic of this paper, primarily focusing on numerical methods for the bidomain model, represents only a small piece of this puzzle. It is, however, an important piece, since cardiac electrophysiology is subject to huge clinical interest and the computation time is a bottleneck for research. Therefore, although the last few years have witnessed remarkable improvements in computational tools for the bidomain model, substantial further progress is required. It is difficult to define a goal for the computation times, but running full scale simulations in real time has been suggested as a natural target [17]. As stated in [7], reaching this level will require effectively combining a number of different techniques, including adaptive discretization methods and efficient utilization of modern, multicore hardware architectures.

References

[1] X. Cai, G. Lines, and A. Tveito. Parallel solution of the bidomain equations with high resolutions. *Proceedings of the Parco03 Conference, Dresden, Germany.* 2003.

[2] R. Leveque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

[3] D. Adams. Propagation of depolarization and repolarization processes in the myocardium - an anisotropic model. *IEEE Trans Biomed Eng,* 38(2):133–141, 1991.

[4] R. L. Winslow, J. Rice, S. Jafri, E. Marban, and B. O'Rourke. Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II, model studies. *Circulation Research,* 84:571–586, 1999.

[5] G. T. Lines. *Simulating the electrical activity of the heart: a bidomain model of the ventricles embedded in a torso.* PhD thesis, Department of Informatics, Faculty of Mathematics and Natural Sciences, University of Oslo, 1999.

[6] M. Potse, B. Dube, J. Richer, A. Vinet, and R. Gulrajani. A comparison of monodomain and bidomain reaction-diffusion models for action potential propagation in the human heart. *Biomedical Engineering, IEEE Transactions on,* 53(12):2425–2435, Dec. 2006.

[7] S. Linge, J. Sundnes, M. Hanslien, G. T. Lines, and A. Tveito. Numerical solution of the bidomain equations. *Philosophical Transactions of the Royal Society A,* 367(1895):1815–2118, 2009.

[8] J. Keener and J. Sneyd. *Mathematical Physiology.* Springer-Verlag, 1998.

[9] U. M. Ascher and L. R. Petzold. *Compuer methods for ordinary differential equations and differential-algebraic equtions.* SIAM, 1998.

[10] J. Sundnes, G. T. Lines, and A. Tveito. Efficient solution of ordinary differential equations modeling electrical activity in cardiac cells. *Mathematical Biosciences,* 172:55–72, 2001.

[11] J. Sundnes, G. Lines, K.-A. Mardal, and A. Tveito. Multigrid block preconditioning for a coupled system of partial differential equations modeling the electrical

activity of the heart. Technical report, Simula Research Laboratory, Norway, 2002.

[12] H. P. Langtangen. *Computational Partial Differential Equations – Numerical Methods and Diffpack Programming.* Springer-Verlag, 1999.

[13] J. Legrice, B. Smaill, L. Chai, S. Edgar, J. Gavin, and P. Hunter. Laminar structure of the heart: ventricular myoctye arrangement and connective tissue architecture in the dog. *Lam. Org. of Ventric. Myocar.*, pages 571–582, 1995.

[14] The visible human project.

[15] H. Osnes, T. Thorvaldsen, S. Wall, J. Sundnes, and A. McCulloch. An operator splitting technique for integrating cardiac electro-mechanics. *Submitted to journal*, 2009.

[16] T. Thorvaldsen, H. Osnes, and J. Sundnes. A mixed finite element formulation for a nonlinear, transversely isotropic material model for the cardiac tissue. *Computer Methods in Biomechanics and Biomedical Engineering*, 8(6):369–379, 2005.

[17] R. Kerckhoffs, S. Healy, T. Usyk, and A. McCulloch. Computational methods for cardiac electromechanics. *Proceedings of the IEEE*, 94(4):769–783, April 2006.

21

A MESSAGE FROM THE HEART

An interview with Per Grøttum and Bjørn Fredrik Nielsen
by Dana Mackenzie

One of the oldest and most reliable diagnostic tools in the cardiologist's arsenal is the electrocardiogram (ECG). The basic technology behind the ECG has not changed in more than six decades. Nine electrodes, placed in specific locations on the torso, measure the electric potential at the surface of the body during the course of a heartbeat.

More than a century of experience and statistical studies have taught cardiologists how to interpret an ECG. For example, physicians can use the change in electric potential between the "plateau" phase and resting phase of the heart cells (called the ST shift) to infer whether a patient has ischemia–in other words, a region of heart muscle that is not receiving a sufficient blood supply. Such regions put the patient at risk for heart attack or myocardial infarction.

There is something very odd about the way ECGs are currently interpreted, though. The ECG is like a message from the heart to the outside world. But instead of trying to understand that message, cardiologists do something more like handwriting analysis. They look at the letters' size or spacing, but they don't actually try to decode what the letters mean.

Bjørn Fredrik Nielsen, of Simula's Scientific Computing department, has been working for the last five years with Per Grøttum of the University of Oslo on a project that promises to radically change the way that ECGs are interpreted. Instead of settling for a graphical analysis of the ECG recordings, they want to translate the message directly. This means solving what mathematicians call an *inverse problem*. Using the measurements of the electric potential at the body's surface, mathematical techniques will allow them to work backwards and figure out what kind of electrical activity in the heart muscle would account for the observed electric fields on

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_21,

the torso. If this inversion can be performed successfully, then the ischemic region should stand out very clearly because it would have a different electric potential from normal heart muscle.

Thus, instead of merely diagnosing that some part of the heart is ischemic–which is all that the 60-year-old technology of today's exercise ECGs can do–the new technique would pinpoint exactly *what* part of the heart is ischemic and the size of the lesion. In addition, Nielsen and Grøttum hope that their approach will improve the sensitivity of exercise ECGs, which currently detect ischemia in only 70 per cent of the patients who have it. Of course, the project is still at a very early phase; Nielsen and Grøttum have so far collected data on 21 patients and performed a complete inversion for three of them.

Nielsen and Grøttum talked with us about their project, about their first clinical success story, and about the ways in which Simula has made this research possible.

"Bjørn, could you tell me how this project got started?"

Bjørn Fredrik Nielsen (BFN): "The inverse problem group at Simula was more or less established for solving this problem. During my PhD studies at the University of Oslo, when I visited the Johann Radon Institute for Computational and Applied Mathematics (RICAM) in Austria, I learned about methods for solving inverse problems. They have been a specialty in Austria; CT scans, for example, are based on integral equations that were originally studied by Johann Radon.

"I used those techniques in my PhD thesis, which I defended in 1997. At that point I was focusing on problems arising in connection with the petroleum industry. Then I worked with the Norwegian Computing Center for three years, basically modeling faults on the Norwegian continental shelf. I was asked to join the team at Simula, and was slowly introduced into the cardiac modeling activity at Simula. In 2003 to 2004, we decided to solve a very practical problem with the use of mathematics and ECG recordings, specifically the problem of trying to compute the physical characteristics of an ischemic region within the heart from ECG measurements."

"What led to the group's decision to study that particular problem?"

BFN: "It was an issue that was already known internationally, and Per had worked on this issue in his doctoral thesis. In 2003 the maturity of the cardiac project was at a level where it made sense to do this in more detail and formulate it as a precise mathematical problem. What was new from our perspective was to look at ischemia as the source for the changes in the ECG."

"Per, what did you do for your PhD, and when was that?"

Per Grøttum (PG): "It's an embarrassingly long story. We were working with infarct-reducing treatments, and we had to have methods of measuring infarct size. We tried to use ECG, or more precisely vectorcardiography[1]. We compared it with en-

[1] This is a primitive technique for spatial analysis of the ECG signal.

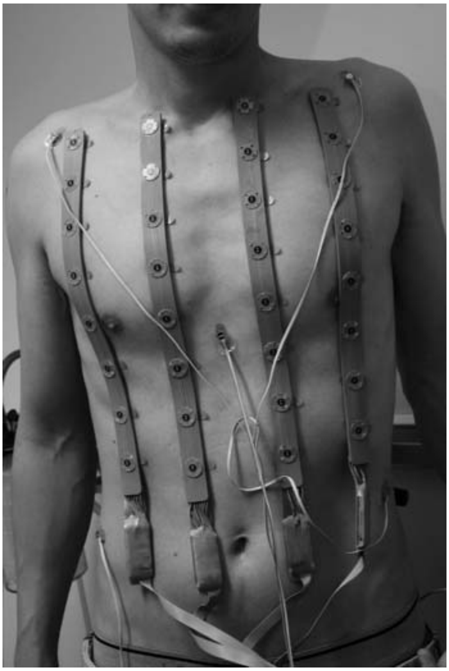


Figure 21.1 ECG equipment with 64 electrodes. Photo: Simula Research Laboratory.

zyme release and also the morphometric size of infarcts, which we measured in dog experiments. Then we applied these methods while using beta blockers during a myocardial infarction, to see if the treatment could save some tissue.

"During that work, it became apparent that we didn't understand enough about the ECG, and the lead system we had, and the vectorcardiography method that was designed in the 1950s from dog experiments. We wanted to see if we could establish a lead system that was better than the standard lead systems, and do the inverse solution more properly to establish what was going on inside the heart. That is what came out of my doctoral work in the 1980s. It was impossible to do anything about that until Aslak Tveito and I met in the 1990s and started to work with the forward problem. During this period we were building up our understanding of the physiology and how to model it, and we thought we might solve the problem with iterative forward solutions. But it was a big step forward when I met Bjørn, with his competence in inverse solutions[2]."

BFN: "We have bought the ECG equipment shown here (Figure 21.1), which currently has 64 electrodes, and we bought eight additional electrodes, four of which

[2] The "forward problem" deals, approximately, with simulating a heartbeat. The "inverse problem" has to do with finding which simulation best matches the laboratory data. Clearly, good "forward" simulations are necessary for solving the inverse problem, but they are not sufficient.

- ECG recording d
- Cost-functional (regularization term q)

$$J_\alpha(\varphi) = \int_{\partial B} (r(\varphi) - d)^2 ds + \alpha q(\varphi)$$

- Inverse problem

$$\min_{\varphi} J_\alpha(\varphi)$$

subject to the constraint

$$\nabla \cdot [(K_i(\varphi) + K_e(\varphi))\nabla r(\varphi) = -\nabla \cdot [K_i(\varphi)\nabla h(\varphi)]$$

Figure 21.2 Mathematical problem

we will place on each side for a total of 72. The patient's ECG is recorded while he or she is exercising on a stationary bike.

"This (Figure 21.2) is the mathematical problem that we solve. We have an ECG recording, and we try to minimize the deviation between the simulated ECG and the recorded ECG, subject to the assumption that the simulated ECG obeys a partial differential equation model where the physical properties of the ischemia enter. So in contrast to statistical methods for analyzing ECGs, we use the laws of nature to model the electric potential in the heart and the torso.

"That's one aspect of our work that is new, the fact that we go back to basic principles to model what is going on in the body and the heart. Standard ECG software is based on statistical methods for analyzing the signal itself."

PG: "It's interesting to note that in the ECG as we know it, the final leads were added in 1942. That means the basic technique, the 12-lead system, hasn't been changed in 65 years. That doesn't mean it's a poor technique, but it's interesting that it has remained unchanged for so long. The way you analyze the ECG is that you look at amplitudes and time periods, for example, how many milliseconds does this interval last. So it's a morphological study. It was developed that way because you didn't have computers in 1942; you just had a ruler and you could measure these things by hand.

"Over the years, an enormous amount of knowledge about these things has amassed. It's based on statistical studies of all these measurements. You end up with a large number of heuristics when you interpret the ECGs. Those heuristics are now implemented in computers. It's the old manmade way of interpreting the ECG that sits in the computer today."

"So it's like having a computer do basic arithmetic".

PG: "Exactly. But with access to modern mathematics and computers, we thought it would be interesting to see if we couldn't get more out of the ECG by processing it in new ways."

Bjørn Fredrik Nielsen and Per Grøttum

BFN: "For the first couple years, until 2006, we basically worked on a series of theoretical issues regarding this mathematical problem and tested it on synthetic data. We achieved good results with the synthetic data, so then we decided to initiate a validation project. It was Per who established the collaboration with Rikshospitalet.

"Patients entering the hospital for cardiac disease are evaluated to decide whether or not they should be enrolled in the project. If yes, they go through three investigations: an MRI scan, an ECG recording during exercise, and finally perfusion scintigraphy. The role of these three modalities is that the MRI scan is used to build a geometrical model of the heart and the body. The ECG is needed to do the inversion. The perfusion scintigraphy is used as a reference, in the sense that it provides us something that we can compare our results with."

"Is that the gold standard for diagnosing ischemia?"

PG: "There is really no gold standard, because the techniques measure different things. If you do angiography you visualize the vessels and you can see the stenotic part. That is frequently used as the gold standard because that can be treated easily[3]. It makes sense to use as the gold standard something that is amenable to treatment.

"However, you can have ischemia without a stenosis in the coronary artery, so you will not detect all ischemias by angiography. Perfusion scintigraphy measures the amount of blood that enters the myocardium, the heart muscle. It allows you to see differences in perfusion. The part of the myocardium that suffers from insufficient blood supply will suffer more during exercise. So you can see a normal perfusion at rest, and then you exercise the patient and you see that the normal myocardium gets a higher perfusion than the ischemic part. It is a differential measurement.

"The ECG is a measure of function. You are looking at the electrical properties inside the cells. When the production of energy in the cell suffers, then the function of the ion channels tends to diminish, and then you get ECG changes.

"So to sum up, the ECG is a functional measure; the scintigraphy tells you the perfusion; and angiography shows you the plumbing. These are three different things. There is really no way to have a gold standard for the truth, because the truth depends on the perspective you have."

"Why are you focusing on the ECG?"

PG: "The three main heart organizations, the American Heart Association, the American College of Cardiology, and the European Society of Cardiology, have published common standards. They all recommend an exercise ECG as the first method of choice for examining patients with angina or when there is a question of ischemic heart disease. It is a simple procedure to perform, it is cheap and it can be done almost everywhere. But the exercise ECG has shortcomings. If you look at

[3] For example, by implanting a stent in the narrowed artery.

big population studies, it has a low sensitivity compared to angiography. The second shortcoming is that from the ECG you can't tell where in the heart the disease sits and you can't tell the volume of the affected part. You have to resort to other types of examination, for instance scintigraphy, to study that. It would be nice to improve the ECG to increase the detection rate and obtain more precise information about the disease."

"Bjørn, in the article you sent me, you wrote about patient 013. Was this the first case where you compared the results to the actual scintigraphy?"

BFN: "Yes, this was the first case that we investigated, in December of 2007. We had this recording on patient 013. Marius Lysaker, a postdoc at Simula who is collaborating in the Inverse Problems project, and I went to a seminar in Berlin and we ran the simulation there. It always appeared that the ischemia in this patient was in the apex, or in ordinary language the bottom of the left ventricle. We were somewhat disappointed, because neither Marius nor I had heard about the possibility of having an ischemia at that point. We tried different simulations, but the computer always said it was at the apex. I sent Per an SMS from Berlin telling him about the 'disappointing' result. (*BFN and PG laugh.*) Then he sent us a message that the apex was the correct position! So we were surprised that the methodology worked for the very first case.

"It turns out that we have now understood more of the problem, and from our perspective it may be that the apex is one of the simpler regions to find using our approach, because it is very close to the body surface. But with traditional ECGs I think it is not necessarily so easy to find this kind of lesion. What do you say, Per?"

PG: "If the patient has a myocardial infarction and you do an ECG during the interventional procedure, you can find out where the disease is in the heart. But that doesn't hold for exercise ECGs, as I said earlier. You tend to get ST segment depression in lead V5, no matter where the ischemia is in the heart."

"Had patient 013 already had a heart attack?"

PG: "Yes. He had an old myocardial infarction, a posterior myocardial infarction. When he exercised on the bicycle, an ischemic region appeared apical to the infarct and spread around the bottom of the heart to the anterior wall."

"So this ischemia was in a different part of the heart from where the infarct was?"

PG: "It was probably the same supply area, but the infarct was an old infarct and the tissue was dead now. So it appeared on the scintigram as an underperfused zone both at rest and during exercise, whereas the ischemia only appeared during exercise."

"Is this patient at risk for a second myocardial infarction?"

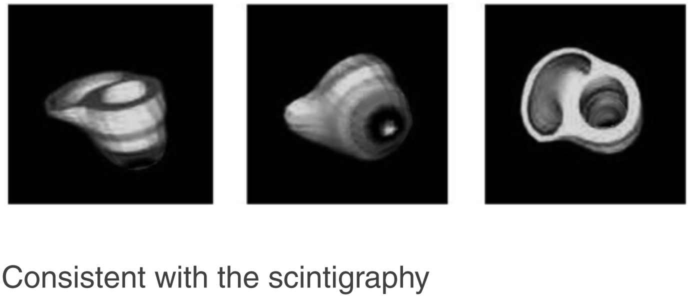


Figure 21.3 Patient 013: Inverse solution

PG: “He went on to angiography, as all of these patients did, and he had a stent put in.”

BFN: “Another surprise to us was that this type of visualization (see figure 21.3), at least for us mathematicians who are not trained as medical cardiologists, is easy to interpret, in the sense that you can easily tell where the left ventricle is and where the right ventricle is.”

“I see that you can even look inside the heart!”

BFN: “And you can zoom in on the location.”

PG: “What Bjørn is saying is important, I think, because the training of medical people today when it comes to interpretation of the ECG is perhaps less than it used to be. So people rely a lot on machines. Having the information that there is a 1 millimeter ST depression in V5 is harder to translate into something useful than an image like this. This image is didactic and easy to interpret. So why not try to do imaging with the ECG, as one does with radiology and scintigraphy and the other modalities?”

BFN: “So one should say that our technique is an imaging technique.”

PG: “It is an imaging of electrical properties.”

BFN: “Mathematically, it’s similar to a computerized tomography (CT) machine. There, you send x-rays through the skull, you measure the decay of the signal as they pass through the skull and you reconstruct a picture. The reconstruction procedure is also based on an inverse problem, not a differential equation in that case but an integral equation. It is different, but it is still an inverse problem. You don’t get a direct image there either, but the image emerges from a mathematical reconstruction.”

"What are some of the challenges that you still have to overcome?"

BFN: "Until recently we have only made qualitative comparison of the scintigraphy with our technique. We are now trying to do a more quantitative comparison. That turns out to be very difficult to do, and it is linked to what Per said, that there is no gold standard. In scintigraphy you have to threshold the images to say if the perfusion is less than a certain amount there is disease, otherwise not. It's the same thing with our technique. We have to use some threshold to estimate the size of the injury. Since there's no gold standard, it's very hard to do this quantitative comparison. That is the main problem we are working on now.

"We also have to do more with the quality of the ECG recordings. They are made during exercise, so a lot of noise enters this recording. Our first solution was to buy some software from SINTEF in Oslo, a noise-removal package."

"Maybe the ECG manufacturers should try to design a machine that can measure people when they're exercising?"

PG: "That's the commercial idea! If we succeed in making a system that can fairly accurately describe the location and extent of a myocardial ischemic area, and do that with a higher sensitivity than it is done today, then I think there will be substantial commercial potential for such a system. Because exercise ECG is so much performed, having a better system is something everyone would want. But it has to be extensively documented, as everything has to be when it comes to medical equipment. Our job is to do the research, and someone else has to take it into industry and make something that can be sold."

BFN: "We are discussing with Simula Innovation how this could potentially be done. Among the main project members, most of us are pure scientists, so the idea of running around in a suit and trying to sell stuff is not what we enjoy."

"You mentioned earlier that the technology had reached a level of maturity so that this project was doable. What were the crucial factors in this maturity?"

BFN: "There are many semi-crucial factors. First, you have to have a group able to do the geometric modeling. Second, you have to have a group that can provide the measurements. Third, you have to have confidence in your signal processing. Finally, you have to have insight into the mathematics of inverse problems. So it was a combination of several factors.

"I think that from a mathematical and also a computing point of view, what we are doing could perhaps have been done during the 1990s. But I think that, very often, applied mathematicians do not want to go into close collaboration with physicians, because the amount of time you need to communicate with people in other scientific fields is so huge."

PG: "It's like digging very deep holes. That is what both of us do, in his field and in my field. The area between the holes doesn't tend to be covered, and we are now trying to cover that."

"Why is Simula a particularly hospitable location for this kind of research?"

BFN: "In Norway, at the universities, it is difficult to have a project working in the way we do. Even though I do not have a lot of power, I can still tell people that we have to do this or that to get closer to our common goal. In Simula, we have this project structure, so that each researcher cannot just go into his office and do whatever he wants. That is perhaps one of the most important reasons why it is possible to do research like this at Simula.

"The other reason is that at Simula we concentrate on long-term research with an applied focus. Also, of course, we benefited from the funding that is available, and furthermore the contact that Simula has with different types of organizations in Norway."

PG: "Another point is that we had the funds to hire Kalkulo to do the geometrical modeling. That is unusual in a university setting, where you don't have the funds to go to a commercial company. You might have the funds to hire doctoral students, and then it might take three to six years before you have the product that we had after a few months."

BFN: "And as soon as the students are trained, they leave, and you have to train new ones!

"In fact, we have bought from both Kalkulo and SINTEF. We have a project with a specific goal. We can take this project and split it into different tasks. We can see that we can handle some of the tasks very well ourselves. Other tasks we could perhaps do, but we do not have the optimal competence for doing them, so it might be better to buy it. In that situation, at Simula, you have the potential to buy rather than do it yourself.

"Also, it must be mentioned that in contrast to some university departments, where interdisciplinary research is not necessarily encouraged, it is encouraged very strongly here at Simula."

"Per, what do people at the hospital think about your project with Simula?"

PG: "They think it's quite interesting, they like the problem, and they realize that mathematics can add something to medicine. Of course, that has been going on for quite a while now. For example, CT wouldn't be possible without mathematics. We work with people in imaging, who are used to computing and modeling, and I think they enjoy it. It's something novel and something that could be very fruitful. We've been very well received at the hospital."

22

CAN ECG RECORDINGS AND MATHEMATICS TELL THE CONDITION OF YOUR HEART?

Bjørn Fredrik Nielsen, Marius Lysaker, Per Grøttum, Kent-André Mardal, Aslak Tveito, Christian Tarrou, Kristina Hermann Haugaa, Andreas Abildgaard, and Jan Gunnar Fjeld

Bjørn Fredrik Nielsen · Marius Lysaker · Per Grøttum · Kent-André Mardal · Aslak Tveito
CBC, Simula Research Laboratory

Christian Tarrou
Kalkulo AS, Norway

Kristina Hermann Haugaa · Andreas Abildgaard · Jan Gunnar Fjeld
Rikshospitalet HF, Oslo, Norway

Bjørn Fredrik Nielsen · Per Grøttum · Kent-André Mardal · Aslak Tveito
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_22,

PROJECT OVERVIEW

Inverse Problems

Heart conditions are among the most widespread and lethal illnesses in the world. For example, it has been estimated that heart infarction was the cause of 18 per cent of the deaths in the United States in 2005. Many people experience poor life quality due to such diseases, and the associated financial costs are enormous.

In 1887, Waller recorded the first electrocardiogram (ECG); a measurement of the electrical potential on the body surface, which is generated by the heart. Today, millions of ECGs are recorded every day around the world. However, despite its apparent success, the traditional human expert-based procedure for interpreting ECG data has its weaknesses; in many cases the procedure simply fails, and manual inspection of the recordings provides rather crude information.

Scientific Challenges

In the Inverse Problems project at Simula, we focus on one particular heart condition, ischemic heart disease. Ischemia is a reversible precursor of heart infarction that typically is caused by (partial) blockage of one or more of the arteries/vessels supplying blood to the heart. Our approach does not follow the methods of contemporary medical research; instead, we use mathematics and computers to gain insight into this health problem. More specifically, we address the following challenge: *Is it possible to use mathematics, medical knowledge, computers, and ECG recordings to determine the size and location of an ischemic region in the human heart?*

A long-term, strategic, and properly funded activity is required to solve this interdisciplinary challenge. The mathematical and software competence available at Simula and our medical connections together make our laboratory an ideal host for the project.

Obtained and expected results

During the first phase of the project we used medical knowledge to develop suitable mathematical models, studied various theoretical aspects, proposed methods, implemented software, and performed computer simulations with synthetic data. These investigations showed that it might be possible to use our approach to identify ischemic heart disease.

In 2006 we bought an ECG device with 64 electrodes, and our collaboration with Rikshospitalet University Hospital was intensified in order to test our schemes using real-world data. Currently, ECGs and geometrical information are recorded from a number of patients at the hospital. We then apply our methods to the data and explore their performance (i.e., determine if they are able to identify the position and size of ischemic regions).

The overall goal of our project is to clarify to what extent it is possible to improve current ECG practises with computational mathematics. Although our results are promising, a number of challenging issues must be addressed to fully explore the potential benefits of the technology.

CAN ECG RECORDINGS AND MATHEMATICS TELL THE CONDITION OF YOUR HEART?

22.1 Introduction

If a coronary artery supplying blood to the heart becomes blocked, the heart will not receive sufficient oxygen, causing an ischemic region. If the condition persists, it will eventually lead to permanent damage, that is, myocardial infarction. Coronary artery disease is one of the most common diseases in the Western world, causing millions of deaths each year. For example, in the United States 18 per cent of deaths in 2005 were due to coronary artery disease [76], while in Denmark around eight per cent of the population experiences poor health because of the disease [77].

The electrocardiogram (ECG) is an important tool for diagnosing and locating ischemic regions. Characteristic changes in the ECG occur when the heart is under stress, as during exercise. This test, however, is highly dependent on the experience and skill of the involved medical personnel. Even when successful, it often provides only a crude picture of the position and size of the affected region [60, 59].

In this study, we explore the possibilities for using mathematical models to compute the characteristics of ischemic regions from ECGs. If feasible, this approach could lead to a new generation of ECG machines.

Ischemic heart disease is usually caused by progressive narrowing of the coronary arteries due to atherosclerosis, which hampers the blood supply to the heart muscle. In the early stages of the disease, the blood supply fails to meet the demand only during vigorous exercise. The imbalance is transient and causes no permanent damage to the heart. In the following we denote this condition as *ischemia*. In the late stages the supply deficit can become permanent and the affected part of the heart muscle then dies, that is, a myocardial (heart) infarct evolves. We denote this condition as *infarction*.

Several tools are available for diagnosing ischemia and infarction. Some, such as angiography and perfusion scintigraphy, assess coronary anatomy and perfusion, that is, the supply side, and the diagnosis is therefore inferential. Others assess the effect of the supply/demand imbalance on function more directly, for example, enzyme leakage, echocardiography, and electrocardiography. Enzyme leakage only takes place in heavily damaged cells and is therefore primarily used to diagnose myocardial infarction.

For decades the ECG has been the primary method to detect ischemia and infarction. It is cheap, non-invasive, repeatable, easy to use, provides immediate results, and reflects the functional condition of the myocardium. The specificity is high, that is, most negatives (healthy patients) are correctly identified. The sensitivity, however, that is, the proportion of actual positives which are correctly identified, is only around 70 per cent [78]. Until now interpretation of the ECG has been qualitative, coarse, and largely based on heuristics. The recording method has been virtually unchanged since its inception around 1940.

Our aim is to detect threatening myocardial infarction in patients admitted to the coronary care unit (CCU) with acute chest pain and to identify significant coronary artery disease in patients undergoing exercise testing. More specifically, we want to improve the ECG by optimising the number of electrodes and the recording positions and by solving the inverse problem, that is, computing the physical characteristics of an ischemic region in the myocardium itself from body surface recordings.

The single most important element in our methodology is the Bidomain equations. This is widely accepted as an accurate model for cardiac electrical activity (see, e.g., [28, 38]). It consists of a system of partial differential equations (PDEs) coupled to a set of ordinary differential equations (ODEs). This complex nonlinear model was introduced during the 1970s and has been thoroughly studied by many researchers (see, e.g. [61, 63, 38, 28] and references therein). The numerical treatment of this system is difficult and even order-optimal schemes require very CPU[1]-intensive software [37, 36, 39]. Hours of computations on high-speed computers are required to simulate the potential distribution changes in the myocardium during a single heart beat. Consequently, from a practical point of view, it is difficult to use the bidomain model in its full complexity to identify ischemic heart disease.

Elevation or depression of the normally isoelectric ST segment in the electrodes overlying the ischemic myocardium is the electrocardiographic hallmark of ischemia. The ST segment changes are due to reduced and altered, but not abolished, electrical activity in the ischemic cells, the so-called transmembrane potential. Nielsen et al. [31] and MacLachlan et al. [32] suggested combining this kind of (a priori) knowledge about the potential distribution in the myocardium, ECG recordings, level set techniques, and one of the equations present in the bidomain framework to define an inverse problem suitable for our application. This approach only involves a scalar PDE and the unknown is the physical domain constituting the ischemic region. Unfortunately, this leads to a nonlinear methodology which requires that the associated forward problem, as well as its adjoint, be solved numerous times in the course of determining the position and size of the ischemic myocardium. This approach is discussed in section 22.3, including numerical experiments with synthetic data.

In general, linear problems are much easier to solve and analyse than nonlinear equations. For example, the level set approach mentioned above leads to relatively time-consuming simulations in 3D. We thus developed a linear methodology for our application [75]. In this formulation we seek to estimate the shift in the transmembrane potential, an important quantity in the bidomain model, from ECG recordings

[1] Central processing unit.

during regional ischemia. Since the transmembrane potential is known to be approximately piecewise constant during specific time intervals of the heart cycle, we can estimate the size and location of the ischemic region from the computed shift in the transmembrane potential. As we will see in section 22.4, this method is derived by neglecting the fact that the intracellular and extracellular conductivities in the heart are altered by ischemia. Thus, additional modelling errors are introduced. On the other hand, this leads to a rather simple parameter identification problem which can be solved and analysed with the methods described in the extensive literature on linear inverse problems. We propose using an all-at-once scheme to solve the associated optimisation problem, that is, a scheme in which one seeks to treat the equation characterising the optimality condition and the potential equation in a fully implicit manner. This leads to a 3×3 linear system of elliptic PDEs.

Our ambitions in this project go beyond gaining further theoretical insights into a relatively mundane mathematical topic; indeed, we seek a solution with real potential value in diagnosinng the ailing heart. Therefore, it has been crucial for this project to obtain real-world data. The collection and interpretation of this data were carried out at Rikshospitalet HF. Initial applications of our method on real-world data had promising results, which will be addressed in section 22.5. At present, this project is at the stage of validating our methods, a grueling process full of new challenges.

In mathematical terms, the task of computing the physical characteristics of an ischemic region from the ECG is an ill-posed problem, because it means that the governing equations do not have a unique solution which depends continuously on the input data. Efficient numerical solution of the associated discrete model is therefore very difficult; the condition numbers of the involved matrices are very large and standard iterative schemes converge slowly. Many scientists have focussed on these challenges and section 22.6 will briefly discuss our contribution to this field. More specifically, for optimality systems arising in connection with rather general linear inverse problems, we have designed an efficient and easily implementable preconditioner. We have proven that the number of iterations needed by our scheme is bounded independently of the mesh parameter Δ and cannot grow any faster than order $O((\ln(\alpha))^2)$, where $0<\alpha \ll 1$ is the Tikhonov regularisation parameter.

Several scientists have analysed inverse problems arising in connection with ECG recordings. In particular, the problem of computing the epicardial potential, that is, the electrical potential at the surface of the heart, from body surface measurements has received much attention [46, 56, 58, 27, 54, 55, 24, 28, 51, 57, 49, 23]. Roughly speaking, the goal of such studies is to compute ECG recordings for the surface of the heart and thereby obtain a deeper understanding of this organ, including signs of its diseases and malfunctions. This task, which we refer to as the *classic inverse ECG problem*, may be formulated as a linear inverse problem for a scalar elliptic PDE. Because of its severely ill-posed nature, it is a difficult problem to solve, but the computational demands are relatively moderate. Furthermore, since it may be expressed as a linear equation, its mathematical properties can be analysed by relatively straightforward techniques. Another frequently studied task in the inverse ECG literature is that of computing the myocardial surface activation wavefront [21, 48, 50, 20, 47].

The mathematical issues regarding the classic inverse ECG problem, including its numerical solution, are well understood and several validation studies have been undertaken [46, 28, 53]. Nevertheless, it is difficult to draw any conclusions from this work. Some of the results reported are promising but it is unknown whether the methods will enter clinical use in the near future or to what extent [46, 53].

The challenge of combining ECG recordings with mathematics and computations to identify ischemic heart disease has not received the same kind of attention. In fact, only a few texts on this topic exist and this research has so far basically addressed the task of revealing the effect of infarctions on ECG signals; in other words, most texts deal with the corresponding direct problem. As far as we know, our work on defining a suitable parameter identification problem for identifying ischemic heart disease is something new and hence untested. This approach leads to a highly nonlinear ill-posed problem that is consequently difficult to analyse and solve. From a practical point of view, however, the following can be noted:

1. In order for the classic inverse ECG problem to be of practical interest, the potential at the heart surface must be computed with a relatively high degree of accuracy. Because of the severely ill-posed nature of the problem, this seems to be extremely difficult.
2. From a medical point of view, only a rather rough description of the size and position of an ischemic region is required. (This is due to the fact that this disease is caused by an occlusion of one or more of the arteries supplying blood to the heart. To treat the illness, we thus only need to know which of the arteries to unblock.) Consequently, the degree of accuracy required in the numerical solution of this problem is moderate and results of practical interest might be possible.
3. Validation of the methodology developed for the classic inverse ECG problem is very difficult. One must somehow simultaneously measure the potential distribution, or activation sequence, at the heart and body surfaces, which requires invasive procedures. Such studies have been performed on patients undergoing catheter ablation surgery by research groups in Innsbruck [26, 1] and Auckland [2, 28]. Promising results have been reported but the required surgery puts severe restrictions on the validation process.
   Because of the large number of persons suffering from ischemic heart disease, it is possible to enroll, within a rather limited time frame, relatively many patients in a validation study of our methods. Furthermore, we can compare our computer simulations with results from, for example, scintigraphy and/or angiography.

Except for the results obtained with real-world data and the discussion of the future of the project in section 22.7, the material presented in this text has already been published in the journal papers [31, 32, 75, 19]. Further results produced within the project can be found in [30, 18, 38, 17].

To summarise, the underlying philosophy of our activities and choices, we want to combine

- mathematical modelling,
- analysis,
- simulation,

- and validation

to solve an important problem, that is, to clearly establish whether one can use mathematics, computers, and ECG recordings to determine the physical characteristics of an ischemic region in the human heart.

22.2 Mathematical Model

The bidomain equations play an important role in our project. This model may be written in the form

$$\frac{\partial s}{\partial t} = F(s,v) \quad \text{in } H, \tag{22.1}$$

$$\chi C v_t + \chi I(s,v) = \nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e) \quad \text{in } H, \tag{22.2}$$

$$\nabla \cdot (M_i \nabla v) + \nabla \cdot ((M_i + M_e)\nabla u_e) = 0 \quad \text{in } H, \tag{22.3}$$

where

- F and I are given functions,
- the exact form of F depends on the cell model in use (see e.g. [28, 38] for details),
- s is a state vector incorporating ionic currents and gating variables,
- v is the transmembrane potential,
- u_e is the extracellular potential,
- M_i and M_e are the intracellular and extracellular conductivity tensors, respectively,
- H is the domain of the heart,
- χ is the area of cell membrane per unit volume,
- and C is the capacitance of the cell membrane.

With the rather limited computing power currently available, it is not possible to use the bidomain model in its full complexity to solve our problem. Not even the new cluster at Simula with 672 cores on 84 computers can provide sufficient computing resources. This is due to the fact that the solution of an inverse problem typically requires that the associated forward model, and in most cases its adjoint, be solved numerous times. The forward problem should therefore be rather cheap/easy to solve but this is by no means the case for (22.1)–(22.3) (see [37, 39, 22]).

We thus need a simplified model. In the present project this has been accomplished by combining the PDE in (22.3) with medical knowledge. More specifically, we exploit the observation that the transmembrane potential v is approximately piecewise constant during certain time intervals of the heart cycle. Furthermore, its properties depend upon whether or not ischemic tissue is present (see [41] for further details).

In mathematical terms, this may be expressed as follows. Let t_1 and t_2 be time instances during the plateau and resting states of the heart cycle, respectively. According to lab measurements,

$$v(x,t_1) \approx \begin{cases} 20\text{mV } x \text{ in healthy tissue,} \\ -20\text{mV } x \text{ in ischemic tissue,} \end{cases} \tag{22.4}$$

and

$$v(x,t_2) \approx \begin{cases} -80\text{mV } x \text{ in healthy tissue,} \\ -70\text{mV } x \text{ in ischemic tissue,} \end{cases} \tag{22.5}$$

(see [6, 5, 4, 68]). From (22.4) and (22.5) we conclude that

$$h(x) = v(x,t_1) - v(x,t_2) \approx \begin{cases} 100\text{mV } x \text{ in healthy tissue,} \\ 50\text{mV} \quad x \text{ in ischemic tissue.} \end{cases} \tag{22.6}$$

Throughout this paper we will refer to h as the shift in the transmembrane potential.

The associated ST shift r in the extracellular potential u_e is defined as the difference between the plateau and resting values of u_e:

$$r(x) = u_e(x,t_1) - u_e(x,t_2). \tag{22.7}$$

(The term ST shift is used because the plateau phase corresponds to the ST segment of an ECG and r represents the shift/change relative to the resting potential.) Since (22.3) must hold for $t = t_1, t_2$, it follows that

$$\nabla \cdot (M_i(x)\nabla v(x,t_1)) + \nabla \cdot ((M_i(x) + M_e(x))\nabla u_e(x,t_1)) = 0 \text{ in } H, \tag{22.8}$$

$$\nabla \cdot (M_i(x)\nabla v(x,t_2)) + \nabla \cdot ((M_i(x) + M_e(x))\nabla u_e(x,t_2)) = 0 \text{ in } H. \tag{22.9}$$

By subtracting (22.9) from (22.8), we conclude that

$$\nabla \cdot ((M_i + M_e)\nabla r) = -\nabla \cdot (M_i \nabla h) \quad \text{in } H, \tag{22.10}$$

where h is defined in (22.6).

The conductivities M_i and M_e will depend on whether or not ischemic tissue is present in the myocardium. Such effects will be included in the level set framework discussed in section 22.3, but will be disregarded in the linear approach considered in section 22.4.

Outside the heart H, that is, in the torso T (see figure 22.1), there are no sources. This means that the potential, and thus the ST shift r, is governed by a standard homogeneous elliptic PDE:

$$\nabla \cdot (M_o \nabla r) = 0 \quad \text{in } T, \tag{22.11}$$

where M_o represents the conductivity in T. Throughout this text we will assume that the body is insulated, leading to the boundary condition

$$(M_o \nabla r) \cdot \mathbf{n_B} = 0 \quad \text{along } \partial B. \tag{22.12}$$

Here, $\mathbf{n_B}$ denotes the outwardly directed normal vector of unit length along the surface ∂B of the body $B = \overline{H} \cup T$. In addition, suitable conditions at the interface between H and T are required:

$$r_H = r_T \quad \text{on } \partial H, \tag{22.13}$$

$$(M_e \nabla r_H) \cdot \mathbf{n_H} = -(M_o \nabla r_T) \cdot \mathbf{n_T} \quad \text{on } \partial H, \tag{22.14}$$

$$(M_i \nabla h + M_i \nabla r_H) \cdot \mathbf{n_H} = 0 \quad \text{on } \partial H, \tag{22.15}$$

(see [38] for a discussion). The normal vectors $\mathbf{n_H}$ and $\mathbf{n_T}$ are depicted in figure 22.1 and we use the notation

$$r_H(x) = \begin{cases} r(x) & \text{for } x \in H, \\ \lim\limits_{y \to x, y \in H} r(x) & \text{for } x \in \partial H, \end{cases}$$

$$r_T(x) = \begin{cases} r(x) & \text{for } x \in T, \\ \lim\limits_{y \to x, y \in T} r(x) & \text{for } x \in \partial H. \end{cases}$$

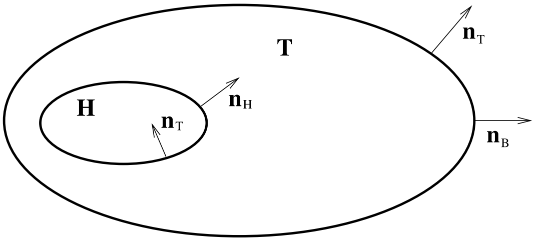


Figure 22.1 A schematic 2D illustration of the involved domains: the heart H, the torso T, and the body $B = \overline{H} \cup T$.

To summarise, the ST shift r is governed by (22.10) in the heart H and by (22.11) in the torso T. Combining these facts with the interface conditions (22.13)–(22.15), the boundary condition (22.12), and Gauss' divergence theorem yields the variational form: Find[2] $r \in H^1(B)$ such that[3]

$$\int_B \nabla \psi \cdot (M \nabla r)\, dx = -\int_H \nabla \psi \cdot (M_i \nabla h)\, dx \quad \text{for all } \psi \in H^1(B), \tag{22.16}$$

where

$$M(x) = \begin{cases} M_i(x) + M_e(x) & \text{for } x \in H, \\ M_o(x) & \text{for } x \in T. \end{cases}$$

If the position, size, and shape of the ischemic region is known, then we can define h according to (22.6) and use (22.16) to simulate the ST shift in the body B. In particular, the effects on the ECG provoked by ischemia in certain regions of the heart can be investigated [68, 66, 64, 65, 67, 62, 69]. In the present context, this is

[2] Here, $H^1(B)$ is the classic Sobolev space of square integrable functions defined on B with square integrable distributional derivatives (see, e.g. [42]).

[3] If r solves (22.16), so does $r + c$ for any constant c. To make the solution unique, we could, for example, add the condition that $\int_B r\, dx = 0$.

the so-called forward, or direct, problem. This is, of course, a very interesting subject in itself and may provide useful information about the mechanisms responsible for ST shifts.

However, we are focussing on the *inverse problem*, that is, to use ST shifts observed in ECG recordings and the model (22.16) to identify ischemic heart disease. Due to the particularly simple structure (22.6) of the shift h in the transmembrane potential, it is clear that the ischemic region can be computed if we manage to recover h from the ECG data. This is basically what we try to do at the Inverse Problems Project at Simula. In the next sections we will consider two different approaches to this problem.

Remarks

1. From a mathematician's point of view, the focus on the shift h in the transmembrane potential and the ST shift r might not be natural. Since the transmembrane potential v possesses a binary property during both the plateau and resting phases of the heart cycle (see equations (22.4) and (22.5)), why not only consider, for exaple, the resting state? Unfortunately, ECG machines cannot measure the potential corresponding to the resting or plateau phases, only the shift on the body surface. Whether or not it is possible to make ECG machines that are more appealing to the theoretical mind is beyond the scope of our project.
2. We want to use model (22.16) for the ST shift r for inverse solution procedures. Input data that correspond to the quantities present in this model are thus required. However, ECG recording devices typically measure potentials relative to the Wilson central terminal[4], not ST shifts. Fortunately, ST shifts can be estimated from ECG data. We will not dwell any further upon this issue, but it is important to note that real world data impose this complication, as well as a series of other challenges, onto our modelling framework.
3. We have not taken the movement of the heart into consideration. One therefore might argue that the derivation of equation (22.10) is incorrect; the heart geometry is different at times t_1 and t_2. We have not investigated the size of the modelling error introduced by this simplification. This should, of course, be done and is on our list of "things to do". Note that this source of error would have been eliminated if the problem mentioned in (1) above were solved.

22.3 Level Set Approach

Let d denote the measured ST shift at the body surface, that is, i.e. $d : \partial B \rightarrow \mathrm{I\!R}$ is a given function representing the available observation data. (In the theoretical part of this text we assume that measurements are available for the entire body surface ∂B. In practice, however, recordings are only made at the positions of the ECG electrodes.) Our goal is to use d and equation (22.16) to recover the position and size of an ischemic region, which we will denote by D, in the heart.

[4] Wilson's central terminal is the arithmetic average of the potentials at the limbs.

From (22.6) it follows that the shift h in the transmembrane potential depends on D. That is, $h = h(D)$ and consequently (22.16) implies that the ST shift also is a function of D; $r = r(D)$. Our scheme for identifying the oxygen-deprived region is based on the output least squares approach. More specifically, we suggest recovering D by minimising the deviation between the observation data d and the simulated ST shift on the body surface. Expressed with mathematical symbols, we may write this problem in the form

$$\min_{D} \frac{1}{2}\|r(D) - d\|^2_{L^2(\partial B)} \tag{22.17}$$

subject to $r(D)$ satisfying

$$\int_B \nabla\psi \cdot (M(D)\nabla r(D))\,dx = -\int_H \nabla\psi \cdot (M_i(D)\nabla h(D))\,dx \quad \text{for all } \psi \in H^1(B), \tag{22.18}$$

where $M = M(D)$ and $M_i = M_i(D)$ are the conductivity tensors which, in general, depend on the presence of ischemic tissue.

In [31] we suggest using a level set function $\phi : H \rightarrow \mathrm{I\!R}$ to represent D:

$$\begin{aligned} \phi(x) &= -\mathrm{dist}(\partial D, x) && \text{if } x \text{ is in } D,\\ \phi(x) &= 0 && \text{if } x \text{ is on } \partial D,\\ \phi(x) &= \mathrm{dist}(\partial D, x) && \text{if } x \text{ is outside } D, \end{aligned} \tag{22.19}$$

where dist($\partial D, x$) is the Euclidean distance between the surface ∂D of the ischemia and x. This means that we employ an implicit representation for D. One can, of course, also apply explicit methods, but we have not investigated whether or not that is beneficial.

Motivated by (22.6), we incorporate the ischemia into h on the right-hand side of (22.18) by the ansatz

$$h(\phi) = 50[1 - G_\tau(\phi)] + 100G_\tau(\phi), \tag{22.20}$$

where

$$G_\tau(r) = \begin{cases} 1 \text{ if } r > \tau, \\ 0 \text{ if } r < -\tau, \\ \frac{1}{2}\left[1 + \frac{r}{\tau} + \frac{1}{\pi}\sin\left(\frac{\pi r}{\tau}\right)\right] \text{ if } |r| \leq \tau \end{cases} \tag{22.21}$$

is a smooth approximation of the Heaviside function[5]. The parameter τ is closely linked to the width of the transition zone between healthy and ischemic tissue and must be estimated from data reported in the biomedical literature [3]. Furthermore, an ansatz similar to (22.20) is used to define a functional relationship between the conductivity tensors and ϕ (see [31] for details).

[5] The Heaviside function function $G : \mathrm{I\!R} \rightarrow \mathrm{I\!R}$ is defined as follows:

$$G(r) = \begin{cases} 0 \text{ for } r < 0, \\ 1 \text{ for } r \geq 0. \end{cases}$$

Since D is implicitly given by the level set function ϕ, we can express (22.17)–(22.18) in terms of ϕ:

$$\min_{\phi} \frac{1}{2} \|r(\phi) - d\|^2_{L^2(\partial B)} \tag{22.22}$$

subject to the constraint

$$\int_B \nabla\psi \cdot (M(\phi)\nabla r(\phi))\, dx = -\int_H \nabla\psi \cdot (M_i(\phi)\nabla h(\phi))\, dx \quad \text{for all } \psi \in H^1(B). \tag{22.23}$$

The numerical treatment of this problem is a difficult subject in itself and certainly beyond the scope of the present text. If one wants to employ a gradient scheme, then one needs to solve an adjoint problem. Details on this matter, including the role of the smoothing parameter τ can be found in [31].

Unfortunately, (22.22)–(22.23) is ill-posed in the sense that ϕ does not depend continuously on the input data d. In mathematical terms, the operator $\phi \to r|_{\partial B}$ is not continuously invertible. We must thus apply some sort of regularisation scheme which yields a family of well-posed approximations of (22.22)–(22.23). One can, for example, employ Tikhonov regularisation or iterative regularisation [45].

In 2006 we developed and tested the model (22.22)–(22.23) with synthetic ECG data. More precisely, synthetic observation data d for the entire body surface were produced in the following manner:

- An ischemic region D was "inserted" into the heart H by changing the cell dynamics and conductivities in the domain occupied by D.
- The electrical activity in the heart and body during a heart beat was simulated with the bidomain model (22.1)–(22.3), using the cell model of Winslow et. al. [40].
- The observation data d were set equal to the ST shift at the body surface ∂B generated by this bidomain simulation.

Thereafter, all information about the ischemic area/volume was put aside and we tried to use the synthetic observation data d and (22.22)–(22.23) to compute the size and position of D. Figure 22.2 shows typical results obtained with this procedure, using iterative regularisation [31, 45]. Please note that the position of the ischemic region is also recovered accurately with (artificial) noise added to d. Similar performance was observed for a wide range of two- and three-dimensional test cases in 2006.

As far as the authors know, the use of level set techniques for solving inverse problems was first suggested by Santosa [33]. Further information about dynamic implicit surfaces can be found in the seminal paper written by Osher and Sethian [34] and in [35].

22.4 Recovering ST Shifts

Consider the mapping $\phi \to r$ implicitly defined by the state equation (22.23). This operator is nonlinear since the conductivity $M(\phi)$ depends on ϕ and the approximate Heaviside function G_τ is used in the ansatz (22.20) for $h(\phi)$. These facts make

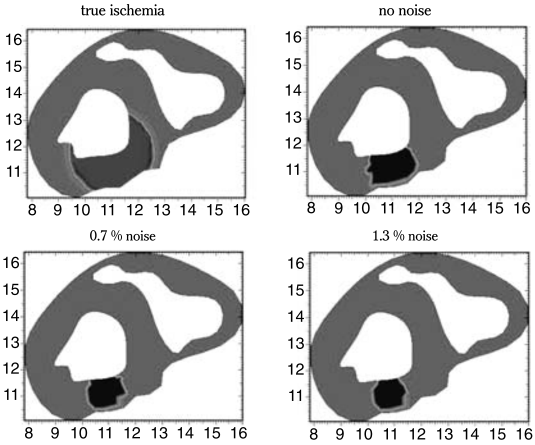


Figure 22.2 The location of the true ischemic region and its estimates computed with noise-free and noisy (synthetic) observation data. Note that the position is recovered rather accurately whereas the size of the lesion is underestimated, an issue that should be further explored. (For the colour version, see figure C.7 on page 646.)

(22.22)–(22.23) relatively hard to analyse and demanding, CPU-wise, to solve. We therefore decided to develop a model involving only linear equations.

Let us now assume that the conductivities M_i and M_e, and consequently M, are not influenced by the presence of ischemic heart disease. More precisely, we accept this modelling error. If $h = h(x)$ is known, then we can use (22.16) to compute the ST shift r throughout the body B. This defines a mapping $h \rightarrow r$ which, under the given assumptions, is easily verified to be linear.

Note that the ischemic tissue D can be estimated from (22.6), that is,

$$D = \{x \in H | \, h(x) \approx 50\}. \tag{22.24}$$

Consequently, if we can use the observation data $d : \partial B \rightarrow \mathbb{R}$ and the linear mapping $h \rightarrow r$ to compute h, then we can approximately recover the ischemic region D with (22.24).

As in the previous section, we propose to use an output least squares formulation to determine h:

$$\min_h \left(\frac{1}{2} \|r(h) - d\|^2_{L^2(\partial B)} + \frac{1}{2}\alpha \|h - 100\|^2_{H^1(H)} \right) \tag{22.25}$$

subject to $r = r(h)$ satisfying

$$\int_B \nabla\psi \cdot (M\nabla r)\, dx = -\int_H \nabla\psi \cdot (M_i \nabla h)\, dx \quad \text{for all } \psi \in H^1(B). \tag{22.26}$$

Here, $1/2\alpha\|h-100\|^2_{H^1(H)}$ is a regularisation term and we employ the notation $r(h)$ to emphasise that the solution r of (22.26) depends on h. This means that we apply the healthy condition 100mV as a (so-called) prior (cf. equation (22.6)). Solving (22.25)–(22.26) is a compromise between minimising the deviation between the simulated and recorded ST shifts on the body surface and determining h such that it is reasonable close to the prior. Unfortunately, (22.25)–(22.26) is severely ill-posed without regularisation and α must therefore be a positive number in practical computations.

Readers not particularly interested in the numerical treatment of (22.25)–(22.26) might want to skip the next two subsections and instead focus on the example presented at the end of this section.

All-at-once Approach

The theory of Lagrange multipliers can be applied in order to write (22.25)–(22.26) in a more convenient form. This yields a 3×3 coupled system of linear PDEs. The details are as follows. Let

$$\begin{aligned} L_\alpha(h,r,w) &= \frac{1}{2}\|r-d\|^2_{L^2(\partial B)} + \frac{1}{2}\alpha\|h-100\|^2_{H^1(H)} \\ &+ \int_B \nabla w\cdot(M\nabla r)\,dx + \int_H \nabla w\cdot(M_i\nabla h)\,dx \end{aligned}$$

be the Lagrangian associated with (22.25)–(22.26). Here, w is the Lagrange multiplier and it is well known (see, e.g. [25]) that the solution (h,r) of (22.25)–(22.26) must be a stationary point of L_α. That is, (h,r) must satisfy the following set of equations:

$$\begin{aligned} \frac{\partial L_\alpha}{\partial h} &= 0, \\ \frac{\partial L_\alpha}{\partial r} &= 0, \\ \frac{\partial L_\alpha}{\partial w} &= 0, \end{aligned}$$

which yields the system

$$\alpha(h,\phi)_{H^1(H)} + \int_H \nabla w\cdot(M_i\nabla\phi) = \alpha(100,\phi)_{H^1(H)} \quad \forall\phi\in H^1(H), \tag{22.27}$$

$$(r,\phi)_{L^2(\partial B)} + \int_B \nabla w\cdot(M\nabla\phi)\,dx = (d,\phi)_{L^2(\partial B)} \quad \forall\phi\in H^1(B), \tag{22.28}$$

$$\int_H \nabla\phi\cdot(M_i\nabla h)\,dx + \int_B \nabla\phi\cdot(M\nabla r)\,dx = 0 \quad \forall\phi\in H^1(B). \tag{22.29}$$

Finally, we can discretise (22.27)–(22.29) with, for example, the finite element method (FEM) to obtain a large linear system of algebraic equations

$$Ax = b. \tag{22.30}$$

No iterative minimisation algorithm is thus needed to determine h and, consequently, an approximation of the ischemic region D (cf. (22.24)). In this approach we aim at simultaneously solving the state equation (22.29), its adjoint (22.28), and equation (22.27), characterising optimality. Such methods are therefore commonly referred to as *all-at-once* schemes.

The ill-posed nature of (22.25)–(22.26) is inherited by (22.30) and the matrix A is ill conditioned for small values of the regularisation parameter α. In addition, the condition number of A grows as the discretisation parameter Δ decreases, making the numerical solution of (22.30) difficult. We will return to these issues in section 22.6.

Transfer Matrix Approach

Rather advanced software is needed for the numerical solution of (22.27)–(22.29). We will therefore consider an alternative approach, commonly referred to as the *transfer matrix method.*

The following notation turns out to be useful. Let R represent the operator that maps h onto its associated body surface ST shift, that is,

$$R(h) = r(h)|_{\partial B}, \tag{22.31}$$

where $r(h)$ denotes the solution of (22.16). Furthermore, we introduce the cost-functional

$$J_\alpha(h) = \frac{1}{2}\|R(h) - d\|^2_{L^2(\partial B)} + \frac{1}{2}\alpha\|h - 100\|^2_{H^1(H)}. \tag{22.32}$$

Assume that h has been discretised in terms of a weighted sum of basis functions:

$$h(x) = \sum_{i=1}^{m} p_i N_i(x), \tag{22.33}$$

where $p_1, p_2, \ldots, p_m$ are scalars and $N_1, N_2, \ldots, N_m$ are prescribed functions defined throughout the heart H. Then we may consider h, r, R, and J_α to be functions of the m real variables $p_1, p_2, \ldots, p_m$:

$$\begin{aligned}
h &= h(p_1, p_2, \ldots, p_m), \\
r &= r(p_1, p_2, \ldots, p_m), \\
R &= R(p_1, p_2, \ldots, p_m), \\
J_\alpha &= J_\alpha(p_1, p_2, \ldots, p_m).
\end{aligned}$$

We can therefore express the discrete approximation of (22.25)–(22.26) in the following form: Find $h(x) = \sum_{i=1}^{m} p_i N_i(x)$ such that

$$J_\alpha(h) = \min_{p_1, p_2, \ldots, p_m} J_\alpha(p_1, p_2, \ldots, p_m). \tag{22.34}$$

Recall that we assume that the conductivities M_i and M_e, and consequently M, are not influenced by the presence of ischemic heart disease. Therefore the solution r of (22.16) depends linearly on h and the operator R, defined in (22.31), is linear. Consequently,

$$R(h) = R\left(\sum_{i=1}^{m} p_i N_i\right) = \sum_{i=1}^{m} p_i R(N_i)$$

and

$$\begin{aligned} J_\alpha(p_1, p_2, \ldots, p_m) &= \frac{1}{2}\|\sum_{i=1}^{m} p_i R(N_i) - d\|^2_{L^2(\partial B)} \\ &+ \frac{1}{2}\alpha\|\sum_{i=1}^{m} p_i N_i - 100\|^2_{H^1(H)}. \end{aligned}$$

Next, the first- order necessary condition for a minimum

$$\frac{\partial J_\alpha}{\partial p_j} = 0 \quad \text{for } j = 1, 2, \ldots, m,$$

yields the system

$$\begin{aligned} &\int_{\partial B}\left(\sum_{i=1}^{m} p_i R(N_i) - d\right) R(N_j)\, dx + \alpha \int_H \left(\sum_{i=1}^{m} p_i N_i - 100\right) N_j\, dx \\ &+\alpha \int_H \left(\sum_{i=1}^{m} p_i \nabla N_i \cdot \nabla N_j\right) dx = 0 \end{aligned}$$

for $j = 1, 2, \ldots, m$, or

$$\begin{aligned} &\sum_{i=1}^{m}\left(p_i \int_{\partial B} R(N_i)R(N_j)\, dx + \alpha p_i \int_H N_i N_j\, dx + \alpha p_i \int_H \nabla N_i \cdot \nabla N_j\, dx\right) \\ &= \int_{\partial B} dR(N_j)\, dx + \alpha \int_H 100 N_j\, dx \end{aligned} \qquad (22.35)$$

for $j = 1, 2, \ldots, m$. This defines a linear system with m equations for the m unknowns $p_1, p_2, \ldots, p_m$. Each of the functions $R(N_i)$ are determined by setting $h = N_i$ in (22.16), solving this equation for $r_i = r(N_i)$, and setting $R(N_i) = r_i|_{\partial B}$. Thus m elliptic PDEs must be solved in order to construct the system (22.35).

Note that (22.35) can be written in the form

$$\mathbf{F}_\alpha \mathbf{p} = \mathbf{d}_\alpha, \qquad (22.36)$$

where $\mathbf{F}_\alpha$ is an $m \times m$ matrix, $\mathbf{p} = (p_1, p_2, \ldots, p_m)^T$, and $\mathbf{d}_\alpha$ is a vector of length m. Systems of this kind can be analysed with the techniques available in the rich literature addressing linear inverse problems [45, 44, 43]. Furthermore, efficient methods for solving (22.36) are readily available. We will not analyse this equation in any detail in this paper.

The shift h in the transmembrane potential is fully characterised by the real numbers $p_1, p_2, \ldots, p_m$ in the discrete case. One therefore might characterise the action of $\mathbf{F}_\alpha$ applied to $\mathbf{p}$ as a transformation of h into the associated body surface ST shift, hence the name transfer matrix or forward operator.

The transfer matrix approach requires that m elliptic problems of the form (22.16), in addition to (22.36), be solved. Consequently, this can be done with standard software for elliptic PDEs and, for example, Matlab. Nevertheless, if m is large, then this procedure can be computationally demanding. In the all-at-once approach only one large system of algebraic equations (22.30) must be solved. The latter is therefore an appealing alternative, but advanced "tail-ordered" software is needed. We will not dwell any further upon this issue; the method to be used simply depends on implementational skills, available manpower, and the need for efficient computations.

An Example

The model (22.25)–(22.26) has been tested on a number of synthetic problems. As for the level set approach, the artificial observation data d were generated by the procedure described in section 22.3. Figure 22.3 shows typical results obtained for a 3D heart in torso test case with transmural anterior ischemia. To investigate the stability of the method with respect to uncertainties in the size of the heart, the inverse problem was also solved with perturbed heart volumes. That is, the heart volume used in the inverse solution was different from that used to produce the synthetic ECG d.

22.5 Validation

Already from the start of the project we emphasised the importance of testing our schemes on real-world data. Our methodology will certainly not have any influence on medical procedures without validation. In December 2006 we therefore bought a multichannel ECG recording device with 64 electrodes (BioSemi, Netherlands) and initiated a clinical feasibility study in association with Rikshospitalet HF.

Briefly, some patients with symptoms of ischemic heart disease that are referred to the hospital for coronary angiography are first subjected to myocardial perfusion scintigraphy during exercise. Based on clinical judgement, we try to recruit for our study those patients with the highest probability of having ischemic heart disease. If the patient consents to participate in the study, the following procedure is carried out.

- Perfusion scintigraphy with the radioisotope technetium-99m is used to obtain images of myocardial perfusion at rest and under exercise. Comparing these images allows one to visually identify regions where myocardial blood flow is reduced under exercise, that is, reversible ischemic. A tomographic recording technique permits a full 3D myocardial volume reconstruction. Examples of the images are

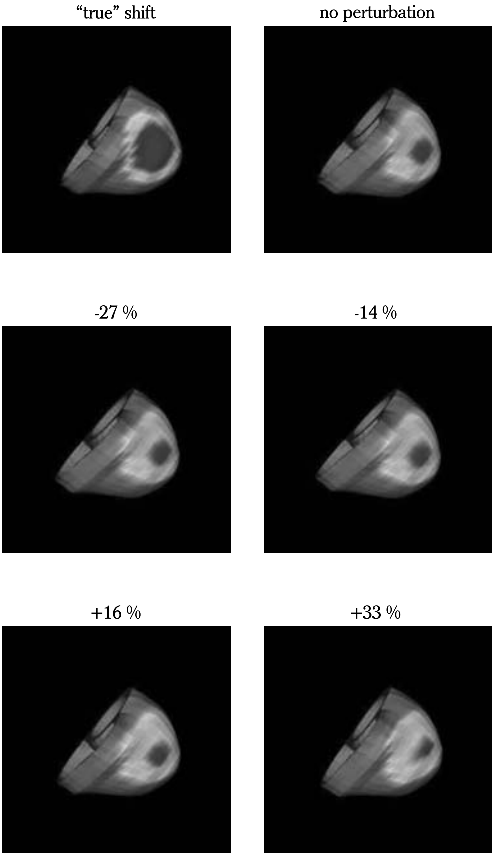


Figure 22.3 Results obtained in 3D for a heart in torso model with synthetic observation data. The figures show the "true" and recovered shifts h in the transmembrane potential. The numbers above the individual panels quantify the volume perturbations of the heart model used in the inverse solution procedure. The size of the heart is scaled with respect to the size of the panels. (For the colour version, see figure C.8 on page 647.)

shown in figure 22.9. We use these data as a golden standard against which we can compare our results. In the early phase of the project we only employed the data qualitatively by visually comparing the scintigraphic images with the results of the inverse electrocardiographic solution. We plan, however, to implement the co-registration of magnetic resonance (MR) and scintigraphic images, segmentation of the ischemic regions, and numerical comparisons of scintigraphic and electrocardiographic data.

- During the exercise a 64-channel ECG is recorded (see figure 22.4). Note that conventional ECGs only comprise 12 leads. Hence, we measure electrical potential at many locations covering a large fraction of the body surface.

- High-resolution MR images are recorded (see figure 22.5). More precisely, we obtain two sets of recordings for each patient:
  - Body images recorded perpendicularly to the longitudinal axis of the body,
  - Heart images recorded perpendicularly to the longitudinal axis of the heart.

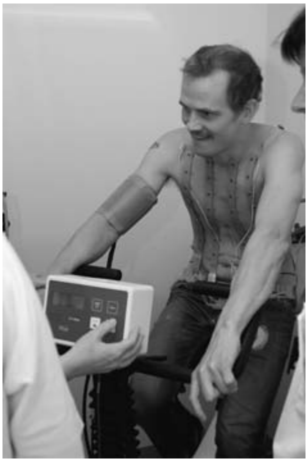


Figure 22.4 The ECG recording system. Photo: Simula Research Laboratory.

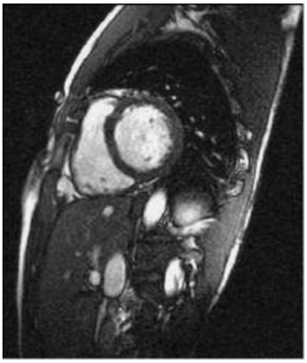


Figure 22.5 An MR image of a cross section of the body.

Patient-specific geometrical models are constructed from the MR data, including finite element meshes suitable for numerical computations (see figure 22.6). This is a rather complicated process, involving automatic and manual segmentation procedures, analysis of 3D curves, lofting techniques, algorithms for handling splines, and tetrahedrisation techniques. More information about this scheme is available in [9]. Furthermore, we use the conductivity values reported in the literature (see, e.g.

[70, 52, 71, 72, 73, 74]), and the fiber structure is modelled by interpolating the measurements for the epicardial and endocardial surfaces presented in [8, 7]. Note that we only model the ventricles and the lungs, that is, the remaining part of the body is treated as an homogeneous medium. The effect of including further organs has not been investigated.

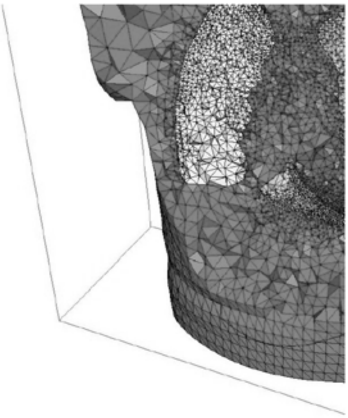


Figure 22.6 A tetrahedrisation of the human torso generated from MR images. In addition to the ventricles, our patient-specific models contain lungs. (For the colour version, see figure C.9 on page 648.)

It is not easy to determine the ST shift d on the body surface, needed in the models (22.22)–(22.23) and (22.25)–(22.26), from the ECG data. This is due to the amount of noise present in the recordings (see figure 22.7). Several schemes for handling this problem are available in the literature and in association with SINTEF we have developed and implemented algorithms for "removing" the noise and estimating d. This is obviously a very critical step in our methodology since the observation data d play such an important role in our (ill-posed/unstable) recovery schemes.

So far we only have data from 16 patients and the results reported in this paper must be regarded as highly preliminary. As an example, we will consider our experience with patient 013. The quality of the ECG recorded from this patient was very good. Only three of the 64 channels had to be excluded due to a very high noise level. Figure 22.8 shows the ischemic region computed based on the inverse problem (22.22)–(22.23). This region coincides with that of perfusion scintigraphy shown in figure 22.9. Please note that the results shown in figure 22.8 were generated without any knowledge about the perfusion scintigraphy data. In fact, since this was the first test we performed with real-world data, we were surprised by the accuracy of the inverse solution.

The results shown in figure 22.8 were produced with a patient-specific geometrical model generated from MR images. From a practical point of view, it is, of course, inconvenient to construct individually tailored models. If one wants to identify is-

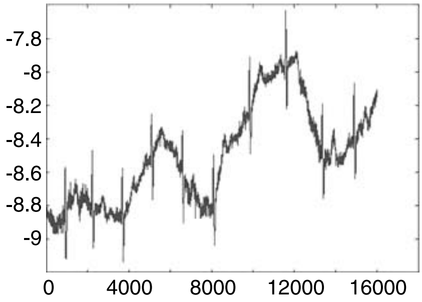


Figure 22.7 A typical ECG recording. The drift and noise must be "removed".

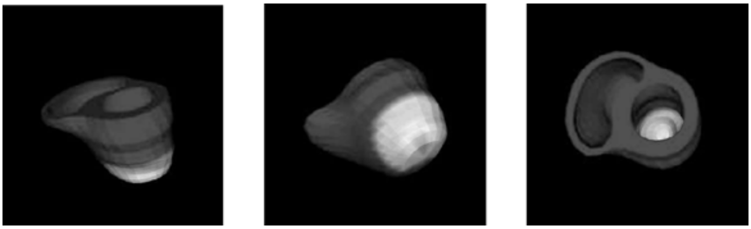


Figure 22.8 The ischemic region computed for patient 013 with our inverse ECG model (22.22)–(22.23). The result is consistent with the scintigraphic images shown in figure 22.9. More precisely, the recovered shift h in the transmembrane potential is shown. Blue indicates healthy tissue ($h \approx$ 100mV) and red ischemic tissue ($h \approx$ 50mV). (For the colour version, see figure C.10 on page 648.)

chemic heart disease efficiently, this step must somehow be avoided; that is, one must conduct simulations on some sort of generic/prototypical tetrahedrisation of the human body. We therefore decided to explore whether our inverse solution procedure is sensitive with respect to changes in the geometry. Figure 22.10 shows an experiment designed to shed some light on this issue. More specifically, inverse solutions were computed with the geometrical models of patients 001, 005, 006, 007, 009, and 013, using the ECG recorded on patient 013. Except for the result obtained on the geometrical model of patient 006, the simulations seem to be rather consistent. Please note that these patients represent a random sample. Their physical characteristics are not particularly similar.

22.6 Numerical Treatment of Optimality Systems

The numerical solution of optimality systems arising in connection with inverse problems is a difficult task. This is due to the fact that the associated linear systems are severely ill conditioned and appropriate values for the regularisation parameters involved must be determined. For example, we are not currently able to solve the linear

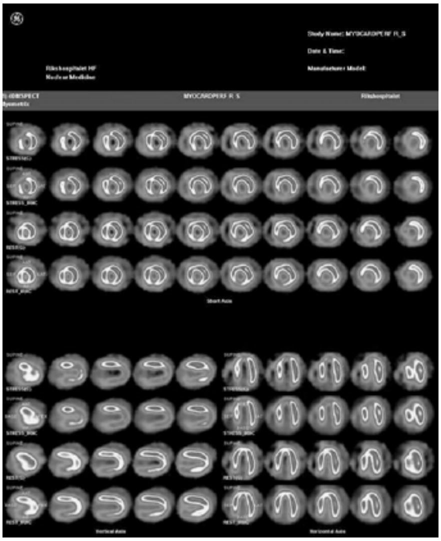


Figure 22.9 A 3D scintigram from patient 013 displaying the uptake of the radioisotope technetium-99m in the left heart chamber. The upper two traces show the uptake in horizontal slices during exercise and the next two traces show the uptake during rest. In the lower right panel the same information is depicted on frontal slices and in the left panel on sagittal slices. By comparing the upper and lower two tracings of the sagittal images, it can be seen that there is less uptake in the right and lower parts of the exercise recordings than at rest, corresponding to reversible inferior and apical ischemia. (For the colour version, see figure C.11 on page 649.)

problem (22.27)–(22.29) in 3D and simultaneously search for an optimal value for the regularisation parameter α on sufficiently fine meshes within reasonable time limits. Solving (22.27)–(22.29) once is not extremely computationally demanding, but this system must typically be solved many times in order to determine a suitable value for α.

Many scientists have proposed and analysed various schemes for this type of problem. In this section we will briefly discuss our contribution to the field. Our paper [19] contains a study, expressed in terms of an abstract framework, of a preconditioning strategy for a broad class of optimality systems. To make the presentation somewhat more concrete, in this text we will limit our discussion to equations (22.27)–(22.29). Even though mathematical proofs are omitted, this section requires that the reader be familiar with functional analysis, variational formulations of PDEs, and the theory of Sobolev spaces.

Recall that H, T, and $B = \overline{H} \cup T$ represent the domains occupied by the heart, torso, and body, respectively. Let $(H^1(H))'$ and $(H^1(B))'$ denote the dual spaces of $H^1(H)$ and $H^1(B)$. For the sake of easy notation, we also introduce the operators

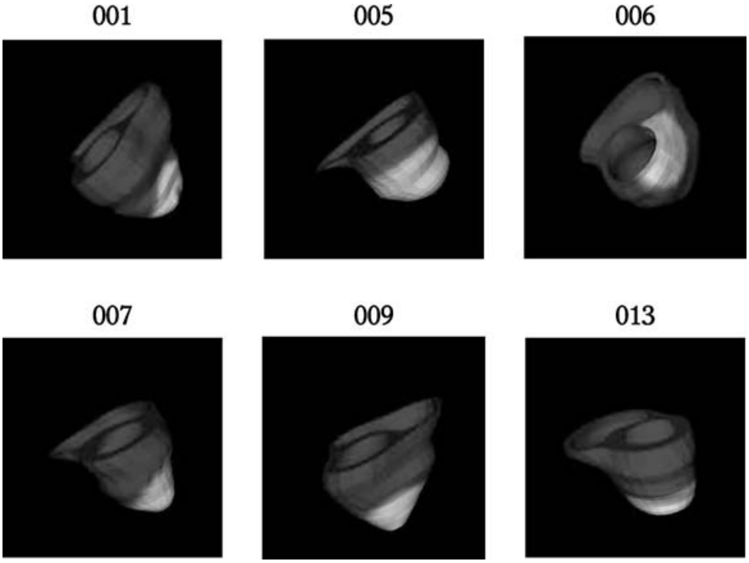


Figure 22.10 Inverse solutions computed with the ECG recorded on patient 013 with different geometries. The numbers above each panel specify which geometry was used to produce the result shown. For example, 001 specifies that the geometrical model of patient 001 was employed. More precisely, the estimated shifts h in the transmembrane potential are shown. Blue indicates healthy tissue ($h \approx 100$mV) and red ischemic tissue ($h \approx 50$mV). (For the colour version, see figure C.12 on page 650.)

$$
\begin{aligned}
&L: H^1(H) \to (H^1(H))', \quad \psi \to (\psi,\phi)_{H^1(H)} \quad \forall \phi \in H^1(H),\\
&B: H^1(H) \to (H^1(B))', \quad \psi \to -\int_H (M_i \nabla \psi)\cdot \nabla \phi \, dx \quad \forall \phi \in H^1(B),\\
&K: H^1(B) \to (H^1(B))', \quad \psi \to (\psi,\phi)_{L^2(\partial B)} \quad \forall \phi \in H^1(B),\\
&A: H^1(B) \to (H^1(B))', \quad \psi \to \int_B (M \nabla \psi)\cdot \nabla \phi \, dx \quad \forall \phi \in H^1(B),\\
&Q: L^2(\partial B) \to (H^1(B))', \quad \psi \to (\psi,\phi)_{L^2(\partial B)} \quad \forall \phi \in H^1(B).
\end{aligned}
$$

Then we may write (22.27)–(22.29) in the form

$$\mathcal{A}_\alpha p = b, \tag{22.37}$$

where

$$\mathcal{A}_\alpha = \begin{bmatrix} \alpha L & 0 & B' \\ 0 & K & A' \\ B & A & 0 \end{bmatrix},$$

$$p = \begin{bmatrix} h \\ r \\ w \end{bmatrix},$$

$$b = \begin{bmatrix} 100\alpha L1 \\ Qd \\ 0 \end{bmatrix}.$$

If (h,r,w) solves (22.27)–(22.29), so does $(h,r,w+c)$ for any constant c. We thus add the additional constraint that the integral of the Lagrange multiplier w is equal to zero. From the Babuška-Brezzi conditions for saddle point problems (see, e.g. [29]), it follows that

$$\mathcal{A}_\alpha : H^1(H) \times H^1(B) \times W \to (H^1(H) \times H^1(B) \times W)',$$

where

$$W = \left\{ \psi \in H^1(B) | \int_B \psi \, dx = 0 \right\} \subset H^1(B),$$

is continuously invertible for every $\alpha{>}0$. Furthermore, equation (22.37) is ill-posed for $\alpha = 0$, reflecting the nature of the underlying inverse problem. We thus expect the numerical solution of (22.37) to be difficult for small values of α.

Discretisation of (22.27)–(22.29) with the FEM yields an approximation of (22.37):

$$\mathcal{A}_{\alpha,\Delta} p_\Delta = b_\Delta, \tag{22.38}$$

where

$$\mathcal{A}_{\alpha,\Delta} : V_\Delta(H) \times V_\Delta(B) \times W_\Delta \to (V_\Delta(H) \times V_\Delta(B) \times W_\Delta)'$$

and

$$V_\Delta(H) \subset H^1(H),\ V_\Delta(B) \subset H^1(B),\ \text{and}\ W_\Delta \subset W$$

are FEM spaces with mesh parameter Δ. Provided that the FEM discretisation is sound, $\mathcal{A}_{\alpha,\Delta}$ will inherit the properties of $\mathcal{A}_\alpha$. More precisely, for a fixed regularisation parameter α, $\mathcal{A}_{\alpha,\Delta}$ is well-behaved as $\Delta \to 0$. (This is in contrast to FEM matrices, which typically are ill-conditioned for small values of Δ, even if α is kept fixed.)

If

$$\mathcal{B}_{\alpha,\Delta} : (V_\Delta(H) \times V_\Delta(B) \times W_\Delta)' \to V_\Delta(H) \times V_\Delta(B) \times W_\Delta$$

is an invertible linear operator, then

$$\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta} : V_\Delta(H) \times V_\Delta(B) \times W_\Delta \to V_\Delta(H) \times V_\Delta(B) \times W_\Delta$$

is a linear mapping and we can solve (22.38) by applying an iterative scheme to

$$\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}p_{\Delta} = \mathcal{B}_{\alpha,\Delta}b_{\Delta}. \tag{22.39}$$

We may regard $\mathcal{B}_{\alpha,\Delta}$ to be a preconditioner and the number of iterations needed to solve (22.39) by, for example, the minimal residual method will depend on the distribution of the eigenvalues of $\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}$ (see, e.g. [15, 16]).

In [19] we suggest using a preconditioner of the form

$$\mathcal{B}_{\alpha}^{-1} = \begin{bmatrix} \alpha L & 0 & 0 \\ 0 & \alpha A + K & 0 \\ 0 & 0 & \frac{1}{\alpha}A \end{bmatrix}. \tag{22.40}$$

More precisely, $\mathcal{B}_{\alpha,\Delta}$ is defined to be the FEM approximation of $\mathcal{B}_{\alpha}$ generated on the same mesh as $\mathcal{A}_{\alpha,\Delta}$. It follows from the Riesz representation theorem that $\mathcal{B}_{\alpha}^{-1}$ is continuously invertible, provided that $\alpha > 0$. This property is typically inherited by FEM approximations and, for fixed α, the operator norms $||\mathcal{B}_{\alpha,\Delta}||$ and $||(\mathcal{B}_{\alpha,\Delta})^{-1}||$ are bounded independently of Δ.

Since both $\mathcal{B}_{\alpha,\Delta}$ and $\mathcal{A}_{\alpha,\Delta}$ are well-behaved as $\Delta \to 0$, one can prove that the spectral condition number of $\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}$ is bounded independently of the mesh parameter Δ (for fixed α). This property is illustrated in Table 22.1, which shows the number of minimal residual iterations required to solve (22.39) on a series of different meshes and for various values of α. It should be mentioned that, since the action of $\mathcal{B}_{\alpha,\Delta}$ is computationally demanding to calculate, in practical simulations we employ multigrid approximations of $\mathcal{B}_{\alpha,\Delta}$.

$l \setminus \alpha$	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}
0	32	40	55	42	25
1	28	36	49	52	24
2	26	30	41	51	26
3	28	28	36	47	32
4	29	28	32	41	41

Table 22.1 The number of minimal residual iterations needed to solve (22.39). Here, l is the refinement level of the grid, that is, the mesh size Δ decreases as l increases.

The role of the regularisation parameter α is more delicate. According to Table 22.2, the spectral condition number of $\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}$ increases significantly as α decreases. Hence, the results presented in Table 22.1 are somewhat surprising, since the number of minimal residual iterations only increases moderately as $\alpha \to 0$. The performance of our preconditioner simply cannot be understood by only considering the spectral condition number of $\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}$. A more detailed analysis of the eigenvalues of the preconditioned operator is needed.

Figure 22.11 shows the absolute value of the eigenvalues of $\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}$ in the case of $\alpha = 10^{-2}$. Note that only a very limited number of eigenvalues are close to zero. This is the key feature for understanding the results presented in Table 22.1.

Axelsson and Lindskog [13, 14] have analysed the performance of the conjugate gradient method applied to problems with this type of spectrum. In [19] we generalise some of their findings to the minimal residual method and prove, in the severely ill-posed case, that the number of iterations needed to solve (22.39) cannot grow faster than order $O((\ln(\alpha))^2)$.

As mentioned at the beginning of this section, we have analysed this approach for a rather large class of linear inverse problems. The results for the abstract framework is as presented for the special case discussed above. Consequently, our preconditioning strategy defines an efficient method for solving optimality systems arising in connection with linear inverse problems. Further details can be found in [19] and references therein.

As far as the authors know, the operator theoretical approach to the preconditioning of well-posed systems of partial differential equations was first suggested by Arnold et al. [10] (see also [11, 12]). Our work may be regarded as an extension of their results to ill-posed cases which involve regularisation.

$l \setminus \alpha$	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}
1	16	108	672	5000	29729
2	16	109	680	5076	40157

Table 22.2 The spectral condition number $\kappa(\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}))$ of $\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}$ for various grid refinement levels l and $\alpha = 1, 10^{-1}, 10^{-2}, 10^{-3}, and 10^{-4}$. The mesh size Δ decreases as l increases.

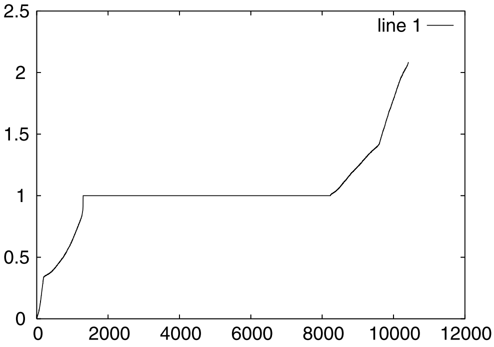


Figure 22.11 Absolute value of the eigenvalues of $\mathcal{B}_{\alpha,\Delta}\mathcal{A}_{\alpha,\Delta}$ with $\alpha = 10^{-2}$.

22.7 Future Perspectives

Even though we had positive results from tests with both synthetic and real-world data, we still do not know whether ischemic heart disease can be identified in terms of an inverse problem for the bidomain model. Further explorations are needed and during the next five years we intend to pursue the following issues:

- A large number of patients suffering from ischemic heart disease must be examined to ensure the validity of the conclusions. Currently, we only have data from two positive cases.
- A healthy test group must be included to establish the specificity of the method, that is, whether our schemes reliably manage to identify healthy hearts.
- The effect of well-known medical factors that may confound ECG interpretations (e.g. bundle branch blocks, electrolyte disturbances) must be explored to establish the specificity of the method.
- The number of electrodes and positions that provide optimal, or at least necessary, sensitivity should be determined.
- The minimal ischemic volume that can be detected should be established, as well as the ability to distinguish subendocardial from transmural lesions.
- More advanced fiber models for the heart should be implemented. The simple interpolation procedure mentioned in section 22.5 is very crude.
- Should other tissues, involving different conductivities, be included in the geometrical model, for example, bone, skeletal muscle, or blood?
- In the derivation of the model (22.16) for the ST shift we ignored the fact that the geometries of the heart during the plateau and resting phases are different. What is the size of this modelling error?
- A thorough sensitivity analysis, including the orientation and fiber structure of the heart, should be undertaken. Which parameters in the model are important and which are not?
- Currently, we only use the measured ST shift d on the body surface in our schemes. Consequently, most of the ECG data are not taken into consideration. Can more information be extracted from the recordings? It is certainly possible to define an output least squares formulation in terms of the complete bidomain model (22.1)–(22.3) and include more of the ECG. This model, however, is extremely computationally demanding and contains a number of parameters that must be determined. In spite of these facts, would such an approach be beneficial? Perhaps one should seek for a model more advanced than (22.16) but not as complicated as (22.1)–(22.3)?
- The noise and drift removal software for the ECGs must be improved.
- The efficiency of the solution of the inverse problems must be improved. Currently, we simulate on coarse meshes. This issue is the main motivation for the mathematical considerations presented in section 22.6.
- The relation to the classic inverse ECG problem mentioned in section 22.1 should be investigated.

The electrical voltage distribution outside the heart is governed by the potential equation (22.11) and the ECG therefore provides a very smooth image of the electrical activity in the heart. Any attempt to extract detailed information about the state of the heart from ECG data is thus bound to lead to a set of unstable equations. This fact must not be forgotten and cannot be circumvented by any mathematical trickery. With this in mind, we might reformulate the purpose of the Inverse Problems Project at Simula to be

> *How much information about the size and position of an ischemic region in the heart can be computed from ECG data?*

The present project involves modelling with PDEs, geometrical modelling, detailed biomedical knowledge, mathematical analysis, numerical analysis, advanced software development, and validation with patient-specific data. This highly interdisciplinary activity therefore requires long-term, well-financed, and thoroughly planned research. The project was initiated in 2003 and we will probably not be able to answer all the questions listed above by 2015. One can therefore easily argue that the Simula system, with its emphasis on strategic, long-term research on problems of genuine value for society, is very important to us. Without Simula, it would have been very difficult to conduct the work presented in this paper.

References

[1] G. Fischer, F. Hanser, B. Pfeifer, M. Seger, C. Hintermüller, R. Modre, B. Tilg, T. Trieb, T. Berger, F. X. Roithinger, and F. Hintringer. A signal processing pipeline for noninvasive imaging of ventricular preexcitation. *Methods of Information in Medicine*, 44(4):508–515, 2005.

[2] L. K. Cheng, G. B. Sands, R. L. French, S. J. Withy, S. P. Wong, M. E. Legget, W. M. Smith, and A. J. Pullan. Rapid construction of a patient-specific torso model from 3D ultrasound for non-invasive imaging of cardiac electrophysiology. *Medical & Biological Engineering & Computing*, 43:325–330, 2005.

[3] M. R. Franz, J. T. Flaherty, E. V. Platia, B. H. Bulkley, and M. L. Weisfeldt. Localization of regional myocardial ischemia by recording of monophasic action potentials. *Circulation*, 69:593–604, 1984.

[4] W. E. Cascio, H. Yang, T. A. Johnson, B. J. Muller-Borer, and J. J. Lemasters. Electrical properties and conduction in reperfused papillary muscle. *Circulation Research*, 89:807–814, 2001.

[5] E. Downar, M. J. Janse, and D. Durrer. The effect of acute coronary artery occlusion on subepicardial transmembrane potentials in the intact porcine heart. *Circulation*, 56:217–224, 1977.

[6] A. G. Kleber, M. J. Janse, F. J. van Capelle, and D. Durrer. Mechanism and time course of S-T and T-Q segment changes during acute regional myocardial ischemia in the pig heart determined by extracellular and intracellular recordings. *Circulation Research*, 42:603–613, 1978.

[7] F. J. Vetter and A. D. McCulloch. Three-dimensional analysis of regional cardiac function: a model of rabbit ventricular anatomy. *Progress in Biophysics & Molecular Biology*, 69:157–183, 1998.

[8] P. M.F. Nielsen, I. J.L. Grice, B. H. Smaill, and P. J. Hunter. Mathematical model of geometry and fibrous structure of the heart. *American Journal of Physiology*, 260:1365–1378, 1991.

[9] B. F. Nielsen, O. M. Lysaker, C. Tarrou, A. Abildgaard, M. MacLachlan, and A. Tveito. On the use of st-segment shifts and mathematical models for identifying ischemic heart disease. *Computers in Cardiology 2005, Lyon, France, September 25-28*, pages 1005–1008. IEEE, 2005.

[10] D. N. Arnold, R. S. Falk, and R. Winther. Preconditioning discrete approximations of the Reissner-Mindlin plate model. *Mathematical Modeling and Numerical Analysis*, 31(4):517–557, 1997.

[11] K.-A. Mardal and R. Winther. Uniform preconditioners for the time dependent Stokes problem. *Numerische Mathematik*, 98(2):305–327, 2004.

[12] E. Haug and R. Winther. A domain embedding preconditioner for the Lagrange multiplier system. *Mathematics of Computation*, 69(229):65–82, 1999.

[13] O. Axelsson and G. Lindskog. On the eigenvalue distribution of a class of preconditioning methods. *Numerische Mathematik*, 48(5):479–498, 1986.

[14] O. Axelsson and G. Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numerische Mathematik*, 48(5):499–523, 1986.

[15] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1994.

[16] A. M. Bruaset. *A Survey of Preconditioned Iterative Methods*, volume 324. Addison-Wesley Longman (currently CRC Press), 1995.

[17] B. F. Nielsen, T. S. Ruud, G. T. Lines, and A. Tveito. Optimal monodomain approximations of the bidomain equations. *Applied Mathematics and Computation*, 184(2):276–290, 2007.

[18] T. S. Ruud, B. F. Nielsen, O. M. Lysaker, and J. Sundnes. A computationally efficient method for determining the size and location of myocardial ischemia. *IEEE Transactions on Biomedical Engineering*, 56(2):263–272, 2009.

[19] B. F. Nielsen and K.-A. Mardal. Efficient preconditioners for optimality systems arising in connection with inverse problems. *Submitted to journal for publication*, 2008.

[20] F. Greensite. Myocardial activation imaging. *Computational inverse problems in electrocardiography*, pages 143–190. WIT Press, 2001.

[21] J. Cuppen and A. van Oosterom. Model studies with inversly calculated isochrones of ventricular depolarization. *IEEE Transactions on Biomedical Engineering*, BME-31:652–659, 1984.

[22] K.-A. Mardal, B. F. Nielsen, X. Cai, and A. Tveito. An order optimal solver for the discretized bidomain equations. *Numerical Linear Algebra with Applications*, 14(2):83–98, 2007.

[23] B. Tilg, G. Fischer, R. Modre, F. Hanser, B. Messnarz, M. Schocke, C. Kremser, T. Berger, F. Hintringer, and F. X. Roithinger. Model-based imaging of cardiac electrical excitation in humans. *IEEE Transactions on Medical Imaging*, 21(9):1031–1039, 2002.

[24] R. Modre, B. Tilg, G. Fischer, and P. Wach. Noninvasive myocardial activation time imaging: a novel inverse algorithm applied to clinical ECG mapping data. *IEEE Transactions on Biomedical Engineering,* 49(10):1153–1161, 2002.

[25] P. Pedregal. *Introduction to optimization.* Springer-Verlag, 2004.

[26] B. Messnarz, M. Seger, R. Modre, G. Fischer, F. Hanser, and B. Tilg. A comparison of noninvasive reconstruction of epicardial versus transmembrane potentials in consideration of the null space. *IEEE Transactions on Biomedical Engineering,* 51(9):1609–1618, 2004.

[27] A. J. Pullan, L. K. Cheng, M. P. Nash, C. P. Bradley, and D. J. Paterson. Noninvasive electrical imaging of the heart: Theory and model development. *Annals of Biomedical Engineering,* 29(10):817–836, 2001.

[28] A. J. Pullan, M. L. Buist, and L. K. Cheng. *Mathematically Modelling the Electrical Activity of the Heart: From Cell to Body Surface and Back.* World Scientific Publishing Company, 2005.

[29] D. Braess. *Finite elements. Theory, fast solvers, and applications in solid mechanics.* Cambridge University Press, second edition, 2001.

[30] O. M. Lysaker and B. F. Nielsen. Towards a level set framework for infarction modeling: An inverse problem. *International Journal of Numerical Analysis and Modeling,* 3(4):377–394, 2006.

[31] B. F. Nielsen, O. M. Lysaker, and A. Tveito. On the use of the resting potential and level set methods for identifying ischemic heart disease; an inverse problem. *Journal of Computational Physics,* 220(2):772–790, 2007.

[32] M. C. MacLachlan, B. F. Nielsen, O. M. Lysaker, and A. Tveito. Computing the size and location of myocardial ischemia using measurements of st-segment shift. *IEEE Transactions on Biomedical Engineering,* 2005.

[33] F. Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM: Control, Optimisation and Calculus of Variations,* 1:17–33, 1996.

[34] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics,* 79:12–49, 1988.

[35] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces,* volume 153 of *Applied Mathematical Sciences.* Springer, 2003.

[36] Z. Qu and A. Garfinkel. An advanced algorithm for solving partial differential equation in cardiac conduction. *IEEE Transactions on Biomedical Engineering,* 46(9):1166–1168, 1999.

[37] K. Skouibine and W. Krassowska. Increasing the computational efficiency of a bidomain model of defibrillation using a time-dependent activating function. *Annals of Biomedical Engineering,* 28:772–780, 2000.

[38] J. Sundnes, G. T. Lines, X. Cai, B. F. Nielsen, K. A. Mardal, and A. Tveito. *Computing the Electrial Activity in the Heart.* Springer-Verlag, 2006.

[39] J. Sundnes, B. F. Nielsen, K. A. Mardal, X. Cai, G. T. Lines, and A. Tveito. On the computational complexity of the bidomain and the monodomain models of electrophysiology. *Annals of Biomedical Engineering,* 34(7):1088–1097, 2006.

[40] R. L. Winslow, J. Rice, S. Jafri, E. Marban, and B. O'Rourke. Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II, model studies. *Circulation Research,* 84:571–586, 1999.

[41] E. Carmeliet. Cardiac ionic currents and acute ischemia: From channels to arrhythmias. *Physiol. Rev.*, 79:917–1017, 1999.

[42] J. T. Marti. *Introduction to Sobolev Spaces and Finite Element Solution of Elliptic Boundary Value Problems.* Academic Press, 1986.

[43] C. W. Groetsch. *Inverse Problems in the Mathematical Sciences.* Vieweg, 1993.

[44] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems.* Springer, 1996.

[45] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Kluwer Academic Publishers, 1996.

[46] *Computational Inverse Problems in Electrocardiography.* WIT Press, 2001.

[47] L. K. Cheng, J. M. Bodely, and J. Pullan. Comparison of potential- and activation-based formulations for the inverse problem of electrocardiology. *IEEE Transactions on Biomedical Engineering,* 50(1):11–22, January 2003.

[48] G. Huiskamp and A. van Oosterom. The depolarization sequence of the human heart surface computed from measured body surface potentials. *IEEE Transactions on Biomedical Engineering,* 35:1047–1058, 1988.

[49] F. Greensite and G. Huiskamp. An improved method for estimating epicardial potentials from the body surface. *IEEE Transactions on Biomedical Engineering,* 45(1):98–104, January 1998.

[50] G. Huiskamp and F. Greensite. A new method for myocardial activation imaging. *IEEE Transactions on Biomedical Engineering,* 44(6):433–446, 1997.

[51] R. S. MacLeod and D. H. Brooks. Recent progress in inverse problems in electrocardiology. *IEEE Engineering in Medicine and Biology,* 17(1):73–83, January 1998.

[52] B. Hopenfeld, J. G. Stinstra, and R. S. Macleod. Mechanism for ST depression associated with contiguous subendocardial ischemia. *Journal of Cardiovascular Electrophysiology,* 15:1200–1206, 2004.

[53] M. P. Nash and A. J. Pullan. Challenges facing validation of noninvasive electrical imaging of the heart. *The Annals of Noninvasive Electrocardiology,* 10(1):73–82, 2005.

[54] R. M. Gulrajani. Forward and inverse problems of electrocardiography. *IEEE Engineering in Medicine and Biology,* 17(5):84–101, September 1998.

[55] O. Dössel. Inverse problem of electro- and magnetocardiography: Review and recent progress. *International Journal of Bioelectromagnetism,* 2(2), 2000.

[56] Y. Rudy and B. J. Messinger-Rapport. The inverse problem in electrocardiography: Solutions in terms of epicardial potentials. *Critical Reviews in Biomedical Engineering,* 16:215–268, 1988.

[57] Y. Rudy and H. S. Oster. The electrocardiographic inverse problem. *Critical Reviews in Biomedical Engineering,* 20:25–45, 1992.

[58] C. P. Franzone, L. Guerri, S. Tentonia, C. Viganotti, and S. Baruffi. A mathematical procedure for solving the inverse potential problem of electrocardio-

graphy: Analysis of the time-space accuracy from in vitro experimental data. *Mathematical Biosciences*, 77:353–396, 1985.

[59] J. Lau, J. P. Ioannidis, E. M. Balk, C. Milch, N. Terrin, P. W. Chew, and D. Salem. Diagnosing acute cardiac ischemia in the emergency department: a systematic review of the accuracy and clinical effect of current technologies. *Ann. Emerg. Med.*, 37:453–460, 2001.

[60] Y. Birnbaum and B. J. Drew. The electrocardiogram in ST elevation acute myocardial infarction: correlation with coronary anatomy and prognosis. *Postgrad. Med. J.*, 79, 2003.

[61] J. Keener and J. Sneyd. *Mathematical Physiology*. Springer-Verlag, 1998.

[62] M. C. MacLachlan, J. Sundnes, and G. T. Lines. Simulation of ST segment changes during subendocardial ischemia using a realistic 3D cardiac geometry. *IEEE Transactions on Biomedical Engineering*, 52(5):799–807, May 2005.

[63] W. T. Miller and D. B. Geselowitz. Simulation studies of the electrocardiogram: II. ischemia and infarction. *Circ. Res.*, 43:315–323, 1978.

[64] P. R. Johnston. A cylindrical model for studying subendocardial ischemia in the left ventricle. *Mathematical Biosciences*, 186(1):43–61, 2003.

[65] P. R. Johnston and D. Kilpatrick. The effect of conductivity values on st segment shift in subendocardial ischaemia. *IEEE Trans. Biomed. Eng.*, 50:150–158, 2003.

[66] P. R. Johnston, D. Kilpatrick, and C. Y. Li. The importance of anisotropy in modeling ST segment shift in subendocardial ischaemia. *IEEE Trans. Biomed. Eng.*, 48:1366–1376, 2001.

[67] D. Kilpatrick, P. R. Johnston, and D. S. Li. Mechanisms of ST change in partial thickness ischemia. *J. Electrocardiol.*, 36:7–12, 2003.

[68] D. Li, C. Y. Li, A. C. Yong, and D. Kilpatrick. Source of electrocardiographic ST changes in subendocardial ischemia. *Circulation Research*, 82:957–970, 1998.

[69] B. Hopenfeld, J. G. Stinstra, and R. S. MacLeod. A mechanism for ST depression associated with contiguous subendocardial ischemia. *J. Cardiovasc. Electrophysiol.*, 15:1200–1206, October 2004.

[70] K. R. Foster and H. P. Schwan. Dielectric properties of tissues and biological materials: a critical review. *Critical Reviews in Biomedical Engineering*, 17:25–104, 1989.

[71] L. Clerc. Directional differences of impulse spread in trabecular muscle from mammalian heart. *The Journal of Physiology*, 255(2):335–346, 1976.

[72] D. E. Roberts and A. M. Scher. Effect of tissue anisotropy on extracellular potential fields in canine myocardium in situ. *Circulation Research*, 50(3):342–351, 1982.

[73] F. Hanser, M. Seger, B. Tilg, R. Modre, G. Fischer, B. Messnarz, F. Hintringer, T. Berger, and F. X. Roithinger. Influence of ischemic and infarcted tissue on the surface potential. *Computers in Cardiology*, 30:789–792, 2003.

[74] P. C. Franzone, L. Guerri, and B. Taccardi. Spread of excitation in a myocardial volume: Simulation studies in a model of anisotropic ventricular muscle activated by point stimulation. *Journal of Cardiovascular Electrophysiology*, 4(2):144–160, 1993.

[75] B. F. Nielsen, X. Cai, and O. M. Lysaker. On the possibility for computing the transmembrane potential in the heart with a one shot method; an inverse problem. *Mathematical Biosciences*, 210(2):523–553, 2007.

[76] H. Kung, D. Hoyert, J. Xu, and S. Murphy. Final data for 2005. National vital statistics reports. Technical Report vol 56 no 10, National Center for Health Statistics, 2005.

[77] Statens Institut for Folkesundhed. Sundheds- og sygelighedsundersøgelsen 2000. Technical report, Statens Institut for Folkesundhed, 2001.

[78] P. Ginn, B. Jamieson, and M. Mendoza. Clinical inquiries. How accurate is the use of ECGs in the diagnosis of myocardial infarct? *The Journal of Family Practice*, 55:539–40, 2006.

23

PAST AND FUTURE PERSPECTIVES ON SCIENTIFIC SOFTWARE

Anders Logg, Hans Petter Langtangen, and Xing Cai

Anders Logg · Hans Petter Langtangen · Xing Cai
CBC, Simula Research Laboratory

Anders Logg · Hans Petter Langtangen · Xing Cai
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_23,

PROJECT OVERVIEW

Computational Middleware

Research in scientific computing relies heavily on the use of appropriate software. Developing such software is very time-consuming, yet integral part, of the research. With better tools for creating scientific software, costs can be reduced and the resulting codes become more robust and reliable. Our goal is to advance scientific software in general and create frameworks for international distribution that can accelerate research in computational science and engineering. The scientific computing group at Simula has a long tradition in advancing programming technologies so it is natural to continue research along this path.

The primary framework at present, FEniCS, is a key tool for our research on numerical methods and various applications as well as an internationally distributed package that attracts significant attention and creates impact.

The software development activity is financed through a Centre of Excellence grant from the Research Council of Norway. According to the proposal for this grant, there are three major research activities, scientific software being one of them and numerical methods and biomedical flows being the other two.

Scientific challenges

The main challenge in the scientific software field is how to deal with the four constraints of *efficiency*, *generality*, *simplicity*, and *reliability*. Historically, these constraints have implied some sort of compromise.

Traditionally, scientific software has consisted of either very specialised hand-coded programs with high efficiency but low generality and reliability or general libraries with high reliability at the cost of reduced efficiency. Most traditional scientific computing codes are also demanding to operate, even for expert users, implying that simplicity is a constraint that is seldom met.

Obtained and expected results

The main results consist of journal papers, books, and, most importantly, free and open software. The primary benefit from this research is the software, which other groups can use to make progress with numerical computations in various scientific fields. The FEniCS software makes it easy to solve a variety of partial differential equations in a very user-friendly and computationally efficient way. Also, ideas in the FEniCS project make it possible to develop software frameworks that obey all four constraints mentioned above. These ideas briefly consist of creating a general and user-friendly program that can automatically generate highly efficient, specialised programs. Hence, we expect this package to make an impact in industry and science.

PAST AND FUTURE PERSPECTIVES ON SCIENTIFIC SOFTWARE

23.1 Introduction

How will scientific software be written in the future? This is a difficult question to answer but the development will be driven by the demands of the users and developers of scientific software. Since the beginning of scientific programming, these demands have been a constant quest for *efficiency*, *generality*, *simplicity*, and *reliability*.

Efficient software is required to solve large problems with limited computational resources or, alternatively, to solve moderate-size problems fast. Often, the solution process for different physical and mathematical problems can share common elements. This can be reflected in general scientific software that can be used to solve large classes of problems. The development of the code should, ideally, be simple in the sense that the required programming work not be significantly greater than writing down the precise mathematical model to be implemented. This principle of simplicity has shown itself to be extremely hard to realise. Finally, scientific codes must be reliable in the sense that the code must be thoroughly verified (i.e., there exists substantial evidence that the mathematical equations are solved correctly). Some also extend reliability to cover validation, that is, the results of the code have been proven to compare well with physical measurements[1].

In the beginning, there was Fortran and very limited computational resources and so naturally the focus was primarily on efficiency and reliability. With limited resources, generality and simplicity were luxuries that could simply not be afforded (let alone fit in memory). Highly efficient code often contains awkward programming constructs and extending the code in a reliable way is challenging. Therefore, once proven reliable, efficient specialised Fortran code tends to have a long lifetime, often decades.

With the development of new and more flexible programming languages, such as C++, accompanied by increasing computational resources, it has been possible to develop more *general* scientific software that can handle more general classes of problems. Generality, however, has a tendency to come with a price of reduced efficiency. On the other hand, generality enhances code *reuse*, that is, the software kernel can be applied to many different problems and tested over and over again by many people, thereby increasing reliability.

[1] Because validation is about solving the right equations and not about programming itself, we restrict reliability to cover verification only in this chapter.

Over the years there has been an increasing focus on *simplicity*. The goal of simplicity is to allow the user to define new problems with a minimum of effort. Graphical user interfaces (GUIs) have indeed made scientific software easier to use but only predefined (preprogrammed) problems can be reached. To solve new problems, one has to either supply some problem-specific code or express the mathematical problem to be solved in a text-based language that mimics the mathematical notation and then let a program automatically generate the necessary code. These two types of simplicity both require generality in the enabling software.

Combining generality, efficiency, simplicity, and reliability is inherently challenging. It is commonly believed that high performance can only be obtained by specialisation, that is, each particular application requires a special-purpose code to run efficiently. This assertion complicates reliability and simplicity. Recent developments in scientific code generation, however, have shown that the assertion is only true in part. Specialised codes are still needed to obtain high performance, but they can be automatically *generated* by a general code whose user interface lives up to the requirements of simplicity.

The scientific computing group at Simula has tried for two decades to understand how to combine generality, efficiency, simplicity, and reliability using various programming languages and styles. This chapter reviews the current state of this understanding and summarises past and present projects on novel techniques for developing scientific software. In particular, we discuss current trends in scientific software, with a particular focus on software for solving partial differential equations (PDEs) by the finite element method. The last part of the chapter describes our ongoing research on the next generation of finite element software.

23.2 A Short History of Programming Languages

In this section, we aim to outline some characteristics of importance to scientific computations with the programming languages Fortran, C, C++, MATLAB®, and Python. These characteristics are often tightly coupled to certain programming styles, a topic we return to in section 23.3.

23.2.1 Fortran

Fortran [31] has been the dominating language for scientific computing for more than half a century, with Fortran 77 [34] probably being the historically most important dialect. In the forthcoming text, we will use the term *Fortran* for the dialects IV, 66, and 77. The more recent versions, Fortran 90/95 and Fortran 2003/2008 [35], will be referred to as *Fortran 90* and *Fortran 2003*. The reason we restrict the short and general term *Fortran* to the old dialects is simply because we believe that the vast collection of Fortran-type software used in scientific computing is written in these dialects.

A fundamental issue for large software packages written in Fortran is the flow of data in subroutines. Two techniques are available for this purpose: holding data in common blocks or shuffling data in and out of subroutines. The variables in common blocks are global and global data are known to be a major source of programming errors. Many packages therefore rely on shuffling data in and out of subroutines. This technique often implies long (and cryptic) lists of subroutine arguments.

For example, say we want to renumber the nodes in a finite element mesh. The minimal representation of a mesh is an array of nodal coordinates and an array of element-to-node associations. In addition, we could need some extra arrays for intermediate storage. Since Fortran cannot dynamically allocate and deallocate memory, the extra storage must be provided as a subroutine argument. The signature of a subroutine for renumbering the nodes in a mesh can be sketched as follows[2]:

```
SUBROUTINE RENMSH(COOR, E2N, NSD, NEL, NNO, NPEL, SCRATCH, ...)
INTEGER NSD, NEL, NNO, NPEL, E2N(NEL, NPEL), SCRATCH(*)
REAL*8 COOR(NNO, NSD)
...
```

Here, the finite element mesh consists of the collection of the variables `COOR` (nodal coordinates), `E2N` (element connectivity), `NSD` (number of space dimensions), `NEL` (number of elements), `NNO` (number of nodes), and `NPEL` (number of nodes per element). The `SCRATCH` array represents the extra required memory in the algorithm.

Subroutine libraries where all the needed data are shuffled in and out of subroutines are easy to combine with other software packages. Any routine can be reused in another context since all data are exchanged by primitive data structures (integers, reals, arrays). On the other hand, if we want to add an argument to the subroutine, the addition must be consistently incorporated in all parts of the library. For example, if the representation of a finite element mesh were to contain an additional array of data, say, information about faces and their neighbours, the new array and its dimension(s) must be included in the definition of the `RENMSH` subroutine, in all of its calls, and in all subroutines called by `RENMSH` where mesh data are transferred as arguments. Many other routines where a finite element mesh is needed will also need to be updated.

An additional challenge is that traditional Fortran compilers have no support for checking the type of subroutine arguments and thereby detecting inconsistencies in declarations and calls. Clearly, the maintenance of software with long lists of subroutine arguments often becomes both time-consuming and error prone.

In the early days of computing it was necessary to utilise the available hardware in an optimal way, frequently resulting in intricate code that was hard to understand, maintain, and extend. This style of programming, where the programmer has a strong focus on optimising all parts of the code, is known as premature optimisation[3] and is unfortunately widely adopted in science. Usually, highly optimal code is needed only in small parts of a scientific application, allowing the possibility of writ-

[2] Originally, Fortran subroutine names could not exceed six characters. We illustrate the challenge of constructing short and descriptive names here. More recent compilers, however, allow longer names.

[3] “Premature optimisation is the root of all evil,” Donald Knuth.

ing clean code and also utilising slower languages than Fortran for the large portions of the application.

Not surprisingly, when Fortran software proves reliable, it gains a firm position among scientists and a long lifetime. Most of the software for mathematical computations used today is written in Fortran, sometimes decades ago (BLAS and LAPACK are primary examples).

Extensions of the Fortran language have brought many important elements from C and C++ into newer Fortran versions. Fortran 90/95 offers the module concept, which can be used to group data and associate operations, much like C++ classes, but without the capability of object-oriented programming in the strict sense. Fortran 2003 extends the language further, with most of the capabilities of C++ in addition to many other interesting features such as co-arrays. The impact of these promising features on scientific computing when compilers become widely available, however, remains to be seen.

23.2.2 C

The C language [30] started to attract attention in the science and engineering communities during the 1980s. It contained a new construct, the *struct*, which enables variables to be grouped together. This construct immediately solved the problem of long argument lists. In the example above, it would be natural to store all variables associated with a finite element mesh in a C struct:

```
struct Mesh
{
    int nsd, nel, nno, npel;
    int** e2n;
    double** coor;
};
```

Then C functions (corresponding to Fortran subroutines) can take a pointer to the struct `Mesh` as an argument. Our renumbering function then appears as follows:

```
void renumber_mesh(Mesh* mesh, ...)
{
    ...
}
```

Moreover, scratch arrays are no longer needed because one can easily allocate and deallocate memory inside a C function (using `malloc` and `free`). The convenience of dynamic memory allocation and deallocation comes at a price: Memory handling has proven itself to be challenging to implement correctly. Strange errors that are hard to track down often arise from incorrect deallocations of memory.

An immediate advantage provided by C in this example is that the `Mesh` struct can be extended with more information, say, about faces and their neighbours, without affecting the definitions of and the calls to functions. All we need to do is add the additional data in the definition of the struct `Mesh`. Hence, it becomes much easier to extend software because errors with incompatible function calls are dramatically reduced. The C compiler also checks that function arguments in calls are of the correct type. Still, it can be non-trivial, even for highly experienced programmers, to code all the structs and pointers correctly in function calls.

Function libraries in C, however, are not so easily combined with other libraries because each library normally applies its own composition of structs. The definition of a struct for a finite element mesh, for example, might be different for different libraries, requiring a translation of structs from one library to another.

23.2.3 C++

During the 1990s, many scientific software developers started to look at C++ [30, 32, 33] as a promising programming language. It extends the idea of the C struct by allowing structs to also have functions operating on the data in the struct. This construct is called a *class* and has been much celebrated in computer science since its invention in the late 1960s in the form of the programming language *Simula* [3]. Today, almost all software systems for business and administration are composed of classes (and usually implemented in languages such as Java or C#, where the programmer is forced to use classes). For a long time, it was believed that C++ and the class construct would deliver significantly lower performance than Fortran and C and this claim was partly true in the early 1990s. As C++ compilers developed, however, several investigators reported efficiency comparable with that of Fortran [17, 18, 15, 14].

As an example, consider again the implementation of a data structure for a finite element mesh, this time in C++. The implementation can look something like this:

```
class Mesh
{
public:
    int num_nodes() const;
    int num_elements() const;

    const int* element_nodes(int element) const;
    const double* node_coordinates(int node) const;

    int local2global(int element, int local_node) const;

    void renumber();
private:
    int nsd, nel, nno, npel;
    int** e2n;
    double** coor;
};
```

The class contains data, represented here as plain arrays, but these are *private*, indicating that users of the class should not access the data directly but, rather, should use the public member functions of the class. For example, instead of looking up the global node number of local node number `i` in element number `e` through the `e2n` array, one calls either `mesh.local2global(e, i)` or `mesh.element_nodes(e)[i]`, if mesh is a variable of type Mesh. The programmer of class Mesh can at any time change the way data are stored, for example, substituting plain C++ arrays by array or list classes. In principle, the change does not affect application code since these access the data through member functions, whose signatures are assumed to remain the same.

In addition to functions for accessing data, classes can also contain functions that operate on and manipulate data. Here, we have placed the function renumber() inside the mesh class such that this function can operate directly on the data structures in the most efficient way. Note that by making the function for renumbering the mesh part of the class, we can rename it from renumber_mesh() to simply renumber() since it is clear from the context that it is the mesh that gets renumbered.

With the class concept in C++, it is possible to build general tools that overcome some of the difficult features of C. For example, one can create 'smart pointers' to make memory allocation and deallocation more automatic and thereby safer. Plain C++ arrays can be wrapped in classes with, for example, checks on index bounds to obtain safer usage. There has been a substantial development of libraries of such handy tools (including STL and Boost), but no standards beyond some general recipes known as design patterns [36].

Just like C, C++ makes it difficult to reuse code from other libraries, because authors of different libraries have constructed different classes, although the underlying mathematical quantities to be reflected by the classes can be the same. To reuse the renumbering algorithm from the library providing the Mesh class, a user must adopt the Mesh class for the storage of meshes. The user can already have his or her own mesh class with a completely different design. For example, it is quite common not to use plain arrays for storage but, rather, operate with classes for nodes and elements. This idea can briefly be sketched as follows:

```
class Mesh2
{
public:
    int num_nodes() const;
    int num_elements() const;
    const Element& element(int element) const;
    const Node& node(int node) const;

    void renumber();
private:
    std::vector<Element> elements;
    std::vector<Node> nodes;
};
```

Here, a Mesh2 consists of a list of Element and Node objects, where an Element is a list of (pointers to) Node objects. The Node class can be defined as follows:

```
class Node
{
public:
    int index() const;
    double x() const;
    double y() const;
    double z() const;
private:
    int _index;        // global node number
    double _x, _y, _z; // nodal coordinates
};
```

Similarly, the `Element` class can be defined as follows:

```
class Element
{
public:
    int index() const;
    const Node& node(int local_node) const;
private:
    int _index;               // global element number
    std::vector<Node*> nodes; // element nodes
};
```

This is a completely different storage structure compared to the original class `Mesh`. If we want to make use of `Mesh`'s `renumber` function in class `Mesh2`, we need to copy the `elements` and `nodes` data structures to the plain arrays needed by the class `Mesh`. One can also notice that the interfaces (i.e., the signatures of the member functions for accessing data) of the two mesh classes are incompatible; application code using class `Mesh` cannot easily switch to class `Mesh2`. A widely used argument for promoting C++ (and classes in general) is that one can replace the private storage structure by a new structure (e.g., by replacing arrays of real numbers and integers by lists of `Element` and `Node` objects) without affecting users of the mesh class. For this to be true, the public interface must be identical in the two classes. Even in this simple example, however, it is impractical to let class `Mesh2` have the same interface as class `Mesh` because it is natural to use `Element` and `Node` objects in the interface to class `Mesh2`.

In general, a programmer who wishes to reuse an algorithm implemented in C++ is almost always forced to adopt new data structures. Sometimes this can be efficiently carried out by moving pointers around but it often implies the expensive copying of data. Standardisation of interfaces and data storage for scientific computing in C++ must therefore be developed to allow for the efficient combination of different libraries, a topic discussed further in section 23.3.6.

We remark that in the mid 1990s, Java appeared as a promising simplified version of C++. Java also attracted quite some interest for scientific computing but never managed to gain substantial popularity and compete with C++. Today, Java plays a minor role in scientific computing.

23.2.4 MATLAB

During the 1990s, MATLAB became increasingly popular among computational scientists. In MATLAB, variables are not declared but are just brought into play when needed. Array operations can be expressed conveniently and compactly, the user has access to a broad selection of highly optimised numerical algorithms, array data can be easily visualised on the fly, and graphical user interfaces can be easily built. Computational scientists simply find themselves more productive when programming in MATLAB compared to programming in traditional languages such as Fortran, C, and C++. Although MATLAB programs typically do not run as fast as programs written in compiled languages, this fact did not prevent MATLAB from becoming very popular throughout all branches of computational science, including the programming of finite element methods for solving PDEs.

As a programming language, MATLAB was originally designed as a more convenient version of Fortran, with array operations, slicing, and a large standard library for numerical computing and visualisation. Features such as classes were later included in a somewhat ad hoc way, at least compared to languages that incorporate the class concept in their original design. Therefore MATLAB stands out as a language with less programming power than, for example, C++. Thus, MATLAB is rarely used for applications other than pure scientific software. For example, no one in their right mind would attempt to implement a web server or word processor in MATLAB.

23.2.5 Python

It would be ideal to combine the convenience of MATLAB with the programming flexibility of C++. This is what the programming language Python offers, in a nutshell. Simple MATLAB scripts look almost the same in Python but the Python language is even more advanced than C++ and the software engineering support in Python also makes the language well suited for large development projects. Python's syntax is very clean, so code snippets frequently look like algorithmic pseudo code[4].

The major concern, however, is the speed of Python loops over array data. Such loops can run 10 to 100 times slower than similar operations coded in Fortran, C, or C++. There are two techniques for speeding up array computations. The first is to use the extension module NumPy to vectorise code (similar to typical MATLAB programming). The second is to migrate computationally intensive parts (mainly loops) to Fortran, C, or C++. Several tools exist for glueing Python with Fortran, C, or C++ code. For example, with F2PY [10] one can (almost) automatically send Python data to subroutines in Fortran or functions in C and get the results back as Python data structures. The authors have performed extensive performance evaluations of Python for scientific computations [28, 24, 29, 25] and demonstrated that good performance can be achieved by the suitable use of vectorisation and migrating critical parts to Fortran, C or C++. One can also use SWIG [9] to generate Python interfaces to C or C++ code. In combination with automated code generation, this process can in fact lead to significant *speedups* compared to handwritten C, C++, or Fortran code. [41, 4].

[4] Many refer to Python as "executable pseudocode".

Python itself is a general-purpose programming language intended for tasks ranging from computer system administration to GUI development and web services. For numerical computing, core Python must be combined with a set of additional packages, for example, NumPy [5] for MATLAB-like array functionality, SciPy [6] for access to rich Fortran/C libraries, and SciTools [8] for MATLAB-like plotting.

23.3 A Short History of Scientific Programming Styles

This section comments on a number of issues relating to programming styles and concepts of importance to scientific software.

23.3.1 Problems With Static Typing

Scientific computations often involve one generic algorithm that can operate on many different data. Functions in statically typed languages like Fortran, C, and C++, however, are always restricted to a specific type of data. For example, think of an iterative solver such as the conjugate gradient method for a linear system which takes a matrix and two vectors as its arguments. In a finite element code, there can be many different ways of storing a matrix, depending on the class of sparsity pattern of the matrix. The matrix will then be stored as a collection of arrays and dimensions. In C and C++ these data are naturally packed together in a struct or a class. Hence, in C++ one will typically define a class for dense matrices, a class for banded matrices, a class for diagonal matrices, a class for structured sparse matrices, a class for general sparse matrices, and so on.

The type of matrix object must be specified in the signature of a linear solver. We therefore end up with several versions of the linear solver, one for each matrix format (`Vector` here is just some class for vectors):

```
conjugate_gradient(const DenseMatrix& A, Vector& x, const Vector& b);
conjugate_gradient(const BandedMatrix& A, Vector& x, const Vector& b);
conjugate_gradient(const SparseMatrix& A, Vector& x, const Vector& b);
```

Inside these functions, the code can be identical. The basic conjugate gradient method only makes use of the matrix to compute matrix-vector products $p = Aq$, which in C++ can read `p = A*q`, `A.mult(q, p)`, or `mult(p, A, q)`.

Ideally, the programmer or user of a linear solver does not want to deal with programming details related to different matrix formats. The same desire arises when there are different classes for different types of meshes, elements, fields over meshes, and so forth. One would thus like to define the linear solver as follows:

```
conjugate_gradient(const Matrix& A, Vector& x, const Vector& b);
```

which accepts *all* the various possible matrix objects for the argument `A`.

In other words, we want to *parametrise types away*. Languages with static typing require us to specify a variable's type. This certainly has advantages and enables compilers to find errors but it restricts a function argument to a particular type, although the code without the type prefix would work with other types. Note that in MATLAB and Python this problem does not come up, since variables and function arguments can refer to any type and the type is never specified. In C++, there are two ways to parametrise types away and overcome the inflexibility of static typing: object-oriented programming and generic (template) programming.

23.3.2 Object-Oriented Programming

The key idea behind object-oriented programming (OOP) is that classes are organised in hierarchies and that an application code can treat all classes in a hierarchy in a uniform way. In our example with different classes for different matrix formats, we can create a superclass `Matrix` at the top of the hierarchy. This class is *virtual* and contains no matrix entries and cannot be used for any computations (it is an *interface specification*). A specific matrix format is then a subclass of `Matrix` and implements the various public functions specified by `Matrix` and a suitable data structure for the matrix format in question. For example, each format class implements an efficient matrix-vector product, utilising the special storage structure of the matrix. We can thus implement subclasses that include `DenseMatrix`, `BandedMatrix`, and `SparseMatrix`. We can then write a solver function where the matrix object is parametrised by the superclass `Matrix`:

```
conjugate_gradient(const Matrix& A, Vector& x, const Vector& b);
```

This general function can be called with any object of any subclass of `Matrix` as argument `A`:

```
BandedMatrix A;
...
conjugate_gradient(A, x, b);
...
SparseMatrix B;
...
conjugate_gradient(B, x, b);
```

In each case, the C++ runtime system detects that in the first call, `A` is of type `BandedMatrix`. Thus, when the matrix-vector product operation is encountered, it is the multiplication operator or function in class `BandedMatrix` that will be called.

With OOP as described, the user of the program can provide input at runtime, resulting in a matrix object of a certain type, say, `SparseMatrix`, and this object can then be passed on to functions just expecting an object of the common type `Matrix`. In the rest of the code one sees only `Matrix` objects and the details of different matrix storage formats are completely hidden.

General libraries where algorithms can operate on different types of data have been made easy with OOP. For finite element programming and many other topics, this enables more flexible software which can more easily be extended by users, compared to what is feasible with subroutine/function libraries in Fortran/C.

There are other important features of class hierarchies besides helping to parametrise types away. Subclasses inherit both data and functions from the parent class, which makes it possible to collect common data and functions in one place and let subclasses reuse that code without copying it. Moreover, classes collected in hierarchies help to organise software into modules that have certain logical dependencies on each other.

23.3.3 Templates and Generic Programming

When calling functions, OOP has some runtime overhead. In the mid- 1990s, C++ was extended with *templates*, which is a direct way of parametrising away types. The common function `conjugate_gradient()` may now be defined as follows:

```
template <typename Matrix>
conjugate_gradient(const Matrix& A, Vector& x, const Vector& b)
{
  ...
}
```

Any object can be fed in as the parameter `A` as long as that object allows the operations required inside the solver function. Matrix classes do not need to be collected in hierarchies, a fact that makes it simpler (than in OOP) for users of a library to develop their own matrix format classes and use them together with the library. Templates have no runtime overhead as OOP has and can therefore potentially lead to more efficient code[5]. On the other hand, a programmer can soon run into intricate debugging with templates.

One can use the C++ template system to write so-called compile-time programs, which has resulted in the development of *expression templates* and the possibility of creating "intelligent libraries" [16, 14]. The technique can be used to, for instance, unroll loops and specialise general code to specific cases. Many have adopted C++ because of expression templates and the possibility of writing general code that is specialised for performance by the template system prior to compilation. A later section argues that this is much more conveniently implemented through code generation in a more expressive language with nicer syntax than C++ templates.

Templates also support another programming technique of significant importance: generic programming. In OOP, algorithms operating on data are usually implemented in the class, that is, tightly connected to the data, whereas in generic programming algorithms and data are separated. Algorithms are implemented in functions where the data types are parametrised by templates. This gives a looser coupling between numerical algorithms and the various data formats. During the last decade this principle has been shown, at least according to our impression, to result in code that is easier to extend than code built on OOP principles. On the other hand, developing with templates typically means that large pieces of code must be written in header files, which can significantly increase compile times and development cycles. This, together with cryptic error messages from compilers when using templates, often makes OOP programming more convenient than template programming.

[5] In the case of the multiplication operator for a sparse matrix, the runtime overhead of a virtual function call can be expected to be small compared to the actual computation of the matrix-vector product.

23.3.4 Python Programming Glues Together Old and New Techniques

Dynamic typing, where variables and function arguments can refer to any object of any type, makes OOP and templates redundant. Python, MATLAB, and several other languages are dynamically typed. Many of the advanced programming techniques associated with OOP and templates are therefore not needed and no problem of parametrising types ever arises. The result is a much simpler application code which is easier to understand for non-specialists in programming.

Classes and structs are very convenient in C++ and C for collecting data of various types into a single variable. Although Python supports the class concept and OOP, one can alternatively use a list or hash structure for collections of data, because the elements in lists and hashes can be mixed and of any type. Functions associated with the data can be standalone functions taking the, say, hash structure as the first argument. In this way, one can simulate the class concept (though without inheritance). The result, according to the authors' experience, is that scientific software based on Python applies classes to a lesser extent than its C++ counterparts.

When Python is chosen as the primary language for building new scientific applications and problem-solving environments, the need for computational efficiency forces the developer to base data structures on (large) Python NumPy arrays rather than collections of many small classes of various user-defined types. In one way, this is a move away from C++-style code back to the old Fortran-style array-based code, but there are two important advantages to this approach. First, integration with other array-based software is very easy. In particular, one can reuse existing highly optimised Fortran and C libraries. Second, if Python loops are migrated to C++ code, that code will often be more efficient when based on arrays and not involving a large number of small user-defined classes. In other words, the nature of numerical Python programming encourages efficient code constructs.

Python-based systems can also use arrays as their basic data structures and at the same time provide more object-oriented views of the underlying array data. If done properly, such designs can combine the convenience of high-level user-defined objects with the efficiency and simple data representation of arrays. Implementations in C++ based on such ideas can achieve three goals: great flexibility, high performance, and ease of integration with external codes. The efficient and flexible mesh class by Logg [27] is an example of this.

23.3.5 General Mathematics and Special Physics

In the first decades of scientific computing, the limitation of computer memory and computational power forced code writers to take advantage of special features in a problem to optimise the code. This created a strong and long-lasting tradition of specialising code to a physical problem, even when the underlying mathematics and numerics could be used to solve a wide range of physical problems. As a consequence, code writers wrote the same basic parts over and over again instead of building on each other's efforts.

A basic example is a finite difference solver for the heat equation in two space dimensions using an implicit scheme in time such that a linear system must be solved at each time level. One straightforward technique consists of winding the finite difference scheme together with an iterative solution method such as Gauss-Seidel's or the successive overrelaxation (SOR) method, resulting in the following algorithm:

$$u_{i,j}^{n,k+1} = u_{i,j}^{n-1} + \frac{\Delta t}{h^2}\left(u_{i-1,j}^{n,k+1} + u_{i,j-1}^{n,k+1} + u_{i+1,j}^{n,k} + u_{i,j+1}^{n,k} - 4u_{i,j}^{n,k}\right).$$

where n denotes the time level, k the iteration number, and i,j is the numbering of the nodes in a regular mesh over the computational domain. Switching to a different iterative method, for example, the symmetric successive overrelaxation (SSOR) method or the conjugate gradient method, requires a complete rewrite of the scheme and the iterative method. A more flexible approach (with greater storage demands) would be to discretise the PDE to create a linear system explicitly as $Ax = b$ and then a call a linear solver such as `conjugate_gradient()` discussed above, or similar implementations of SOR or SSOR, with A and b as input. This principle separates the general software for linear systems from the application-specific PDE and its discretisation.

In the field of finite elements, one has traditionally spoken of a heat conduction element, a porous media flow element, a potential inviscid flow element, a torsion element, and so forth, although all of these are identical from a mathematical point of view and discretise the same equation $-\nabla^2 u = f$ (Poisson's equation). By separating the geometric shape of an element, together with a set of basis functions and degrees of freedom, from the PDEs being solved, one can write one piece of code and apply it to many different PDEs and physical problems. This approach gained substantial popularity in parallel with adopting more flexible programming styles and the C++ language during the 1990s.

Today, we claim that the natural way to design a finite element package is to parametrise over the variational problem $a(v,u) = L(v)$ and not tie the implementation to a particular physical problem or even a set of predefined models. We will discuss this issue extensively. Most of the leading commercial finite element packages (e.g., ABAQUS, NASTRAN, and ANSYS), however, have not yet adopted this idea, although they claim to be generally applicable to a large class of problems.

23.3.6 Heterogeneous Software Environments

Another feature of the first decades of scientific computing was that software packages tended to be self-contained, meaning that the code writers produced all the necessary code needed for problem solving. For example, finite element software would contain pre- and postprocessors, as well as quite general routines for solving linear and nonlinear systems of equations. The idea of reusing such pieces from an existing library to save programming efforts and increase reliability did not receive substantial attention until recently.

Netlib [23] was the first widespread attempt, emerging in the 1980s, to share reusable subroutine libraries between researchers. In the Internet age, many free/open-source projects have managed to attract attention among users and developers to become mature and rich in functionality. Solving a scientific problem today can naturally be done by building on existing relevant libraries and only writing the strictly necessary new code. As the newer libraries are often general and provide a framework for a large class of problems, the philosophy of creating general instead of specialised code also influences application developers.

At the time of this writing, there are a number of well-designed and well-tested publicly available packages of relevance for scientific computations: PETSc, Trilinos, ACTS, Blitz++, uBLAS, MTL4, and GLAS are examples from linear algebra; deal.II, Cactus, Sundance, Getfem++, LibMesh, FreeFEM, GetDP, and FEniCS are examples from finite element computing; VisIt, VTK, OpenDX, gnuplot, Matplotlib, VPython, Viper, and ParaView are examples from visualisation; and NETGEN, TetGen, GRUMMP, and Triangle are examples of meshing libraries. The modern software developer is relieved from reinventing the wheel but must spend more time on searching for existing packages, testing them, and deciding what to use. This is a highly challenging process. Because different packages have completely different application programming interfaces (APIs), one package cannot easily be substituted by another. There is hence a need for standardised interfaces for linear algebra, finite element computing, visualisation, and so forth, if we want to enhance the coding efficiency of new applications and frameworks.

Unfortunately, there has been little progress with respect to the standardisation of interfaces. This problem becomes particularly evident if one looks at C++ libraries for linear algebra. In 2008, 29 years after the introduction of BLAS, there is still no standard for linear algebra in C++. Several sophisticated libraries exist (e.g., PETSc and Trilinos), but each such library implements its own data structures and algorithms such as linear solvers are tied to the chosen data structures. The authors of the present chapter have been involved with defining common (minimal) interfaces for finite element assembly [26] and MATLAB-style plotting [8], with other interface definitions in progress (for linear algebra and variational forms). Such interfaces can greatly ease the combination of different libraries, if the libraries provide a thin layer implementing the interface.

23.3.7 Diffpack

Many at the Simula Scientific Computing department gained extensive experience with developing software for PDEs through the Diffpack project [11, 37] in the 1990s. Diffpack was started in 1991 and aimed to explore the use of C++ and classes for solving PDEs, which at that time was a highly immature topic. The project built on experience from coding a large, general finite element library in Fortran 77 [12]. The question was how to utilise classes to increase generality, reliability, and simplicity with only a small loss of efficiency.

Arrays and Fortran/C-style loops were chosen to implement all core numerics, with suitable wrappings of arrays in C++ classes for vectors and matrices of various formats. Intensive use of classes appeared at an abstraction level above the core numerical algorithms, mainly to administer the choice of data structures and algorithms, and OOP was extensively used to parameterise types away to achieve programming flexibility. Typical examples of class hierarchies found in Diffpack are vectors, matrices, linear systems, linear solvers, preconditioners, nonlinear solvers, meshes, fields over meshes, finite elements, probability densities, random number generators, and databases with computed fields. Throughout the 1990s, several other C++ packages for PDEs emerged. Although these were mostly developed independently of each other, their designs appeared to be quite similar. The major difference was in how the core numerics were implemented, either via Fortran/C-style loops over arrays or via objects (classes `Mesh` and `Mesh2` from section 23.2.3 highlight this difference).

In the design of Diffpack, special emphasis was made to separate the general mathematics from the particular physics of a problem (see section 23.3.5). The Diffpack library contains a wide range of general algorithms operating on a wide range of data structures. The application programmer writes a fairly short code specifying the physical and numerical parameters in the problem and, in the case of finite elements, a function that samples the integrand in the variational formulation at an integration point in an element. Most of the finite element engine is identical from problem to problem and hence implemented in the general library. User-defined "slots" or callbacks in the overall solution algorithm, such as integrands in the variational formulation, are virtual functions supplied by the application programmer.

Diffpack was initially designed without any attention to parallel computing, multilevel methods (multigrid, domain decomposition), and adaptive mesh refinements. In the beginning of the 1990s it was widely argued that such features had to be taken into account while designing the very basic classes of C++ libraries. Adding the features later, "on top" of a library was believed to introduce significant performance loss. Despite this knowledge, parallel computing [50, 51, 52], multilevel methods [53, 54], adaptive mesh refinement [37], mixed finite elements [55], and support for stochastic PDEs [56] were added as managing classes on top of the original Diffpack software. Extensive performance results (many available in the cited references) showed that efficiency loss was negligible. We believe that these add-on modules demonstrate that it is possible to extend "simple" serial libraries for basic numerical methods with more advanced methods in a software layer above and still obtain decent performance. Today, it seems to be common practice *not* to wire parallel computing, multilevel methods, and adaptive meshes into the most basic classes in numerical PDE libraries.

Publicly available versions of Diffpack were released in 1995 and 1997, resulting in thousands of downloads and quite some attention in this early stage of the World Wide Web. Diffpack went commercial in 1997 through the company Numerical Objects AS. The technology was later sold to the German company inuTech GmbH in 2003, which continues to support the package. Diffpack's customer list includes companies such as Daimler-Chrysler, Intel, Mitsubishi, Nestlé, Siemens, Xerox, and NASA, and universities such as Cambridge, Harvard, Cornell, and Stanford. The package has been applied to solve PDEs in many areas of science and engineering, including physics, geoscience, mechanical/marine/civil engineering, telecommunication, medicine, and finance.

Diffpack features a novel system that lets each class define information about its input data. This information can be added to a tree structure reflecting an input menu. The additions are made recursively, so with just a few lines in the application code one can obtain a comprehensive input menu that can be operated through a GUI, command-line arguments, or a simple command language in a file. This system also allows multiple values to be specified as input for a menu item, resulting in a set of simulations over all combinations of all items with multiple values. Besides the C++ library, Diffpack also contains a large set of Bash, Perl, and Python scripts for file and directory handling, visualisation, movie making, generating tailored GUIs, and man page lookups, to mention some features. The total set of software makes Diffpack a comprehensive problem-solving environment (PSE) for conducting science in an efficient and reliable way and this fact was perhaps the main reason for Diffpack's popularity.

Creating new Diffpack applications seldom requires more than two to five pages of code but writing this code demands expert knowledge of the finite element method and fluency in C++, since the initial goal of the package was to relieve the expert from writing code that could be implemented once and for all in a general library. There were major plans for redesigning Diffpack with templates (see section 23.3.3) and for automatically generating the main portion of C++ code when creating an application (see section 23.6.3). The large number of users requiring backward compatibility, however, prevented these plans from being realised.

As a numerical package, Diffpack was self-contained with its own linear algebra package, vectors, matrices, meshes, finite elements, and so on. Only mesh generation and visualisation were left to third-party applications. Most other C++ packages from the 1990s followed the same trend, resulting in people writing the same type of code over and over again. Many more specialised packages have grown to a very mature state over the last few years, so today it is more natural to build new finite element frameworks on top of linear algebra packages, finite element libraries, and so on, as mentioned in section 23.3.6. This is exactly what we do in the FEniCS project, to be described in section 23.7.

Finally, this section will comment upon how successful Diffpack is in addressing the four software requirements of generality, efficiency, simplicity, and reliability. It has been demonstrated through the numerous Diffpack applications that the package can be used to solve a wide range of PDEs throughout science and engineering, which proves generality. Since the same numerical engine is applied in a large number of various applications, Diffpack has proved to be successful with respect to code reuse and hence reliability. The implementation of general (i.e., widely applicable) numerical algorithms obviously makes Diffpack less efficient than highly specialised codes. By slightly modifying an existing standard Diffpack application, however, it was shown that the efficiency of that application could compete with specialised Fortran code for some important PDEs [17].

Whether Diffpack displays simplicity is more questionable. At the time Diffpack was released to the public in 1995, the efforts needed to solve a new PDE were significantly less than what researchers were used to with Fortran and C libraries. Nevertheless, an essential piece of C++ code is necessary to solve a PDE problem in Diffpack and similar C++ libraries. This code does not closely resemble the simplest mathematical specification of the problem in question. The FEniCS project aims much higher in the quest for simplicity. In addition, the FEniCS software has more ambitious goals than Diffpack for efficiency *and* generality, as we explain in sections 23.6 and 23.7.

23.4 Current and Future Trends

The most important advances of scientific software have historically been related to new programming styles and languages, separation of the mathematics (numerics) and the physics of a problem, and utilisation of other people's codes. All of these advances seem very natural in hindsight; in fact, they mirror a similar evolution of physics and mathematics over the centuries. It is difficult to predict the future, but we elaborate below on two important trends. One is driven by recent developments in hardware and the other is driven by recent developments in scientific software.

Recent developments in hardware architecture are currently making parallel computing mandatory. It is no longer possible to scale up the CPU frequency with each new generation of CPUs. Instead, more and more cores are bundled into each CPU and more and more CPUs are bundled onto a single motherboard. One can foresee cheap laptops with hundreds of cores and mainstream number-crunching clusters consisting of hundreds of PCs, each with hundreds or thousands of cores. How to efficiently program such architectures remains an open question, but for the application developer it is important that programming environments offer abstraction layers that hide the low-level parallel code from the higher-level algorithms used to solve the scientific problem.

Recent developments in programming languages have led to an increased focus on automation. It is important not only that scientific software run fast but it must also be fast to develop. By automating key tasks in the implementation of scientific software and ultimately automating the whole process, one can speed up design cycles and shift the focus from programming to modelling. At the same time, one can minimise sources of errors by removing error-prone and tedious tasks. We thus predict that parallel computing and automation will be two major trends for scientific software in the future. We elaborate on these topics below.

23.5 Parallel Computing

Parallel computing is not a new invention. The concept of supercomputing, the use of multiple functional units, and fork & join operations were introduced as early as in the 1950s (see, e.g., [57]). Since then, both parallel hardware and programming paradigms have undergone tremendous changes. Despite its long history, parallel computing has always been a niche technique used by a relatively small number of researchers until quite recently. Limited access to parallel computers, which used to be very expensive, is the main reason. The difficulty in parallel programming itself, however, is another reason, although heroic efforts have been made to develop specialised parallel languages and/or compilers. Since programming tools are predicted to have trouble following today's parallel hardware development [58], attention must be given to the user friendliness of parallel scientific software. This section thus aims to shed some light on the topic of developing user-friendly parallel software while ensuring high parallel performance.

23.5.1 Hardware Considerations

The mass production of processors with multiple cores is the main reason for the 'revival' of parallel computing in recent years. Supercomputers nowadays are simply big clusters of compute nodes that have several multicore processors (see figure 23.1). This is evident from the famous TOP500 list [59].

The hardware buildup of today's parallel systems has a hierarchical layout. On the top level, the compute nodes do not share the computing resources except for an interconnect network between them. Internode communication is typically of the form of sending and receiving messages, for which the message-passing interface [60] (MPI) is the de facto programming standard. On the middle level, memory is shared between the processors inside one compute node. Consequently, intranodal communication is either in an explicit form of message passing, which can now be carried out by direct memory copy, or in an implicit form via threads programming, such as provided by the OpenMP [61] standard. On the bottom level, the cores on one processor chip not only share the entire memory of the compute node but can also share one level of cache. Communication on this level thus has the potential of being (much) faster than on the two other levels [63]. Such a hierarchical hardware architecture is expected to prevail at least in the near future, with more (and simpler) cores being added to a single processor [62]. Another upcoming trend is the addition of special hardware units, called accelerators, to multicore architecture to further improve parallel computing capacity. Current examples of such accelerators are general-purpose graphic processing units, field-programmable gate arrays, and the synergistic processor elements found in the cell broadband engine processors. Programming heterogeneous multicore hardware will be even more challenging. Therefore, the architectural feature of hierarchy (and heterogeneity) must be taken into account when developing parallel software.

23.5.2 Software Development

For scientific computations, parallelism arises from dividing the entire computational work into concurrent pieces. In the case of solving PDEs, for instance, work division typically translates to domain/mesh partitioning. For distributed-memory systems, the best implementation approach is to avoid building up global data structures and, instead, let each unit have its own local data structure. The global data structures are thus distributed and explicit message passing (using MPI) is typically needed when conducting the computations in parallel. Moreover, if an imbalance of work division arises due to some dynamic feature of the computations, shuffling work between neighbours can be considered. For shared-memory systems, on the other

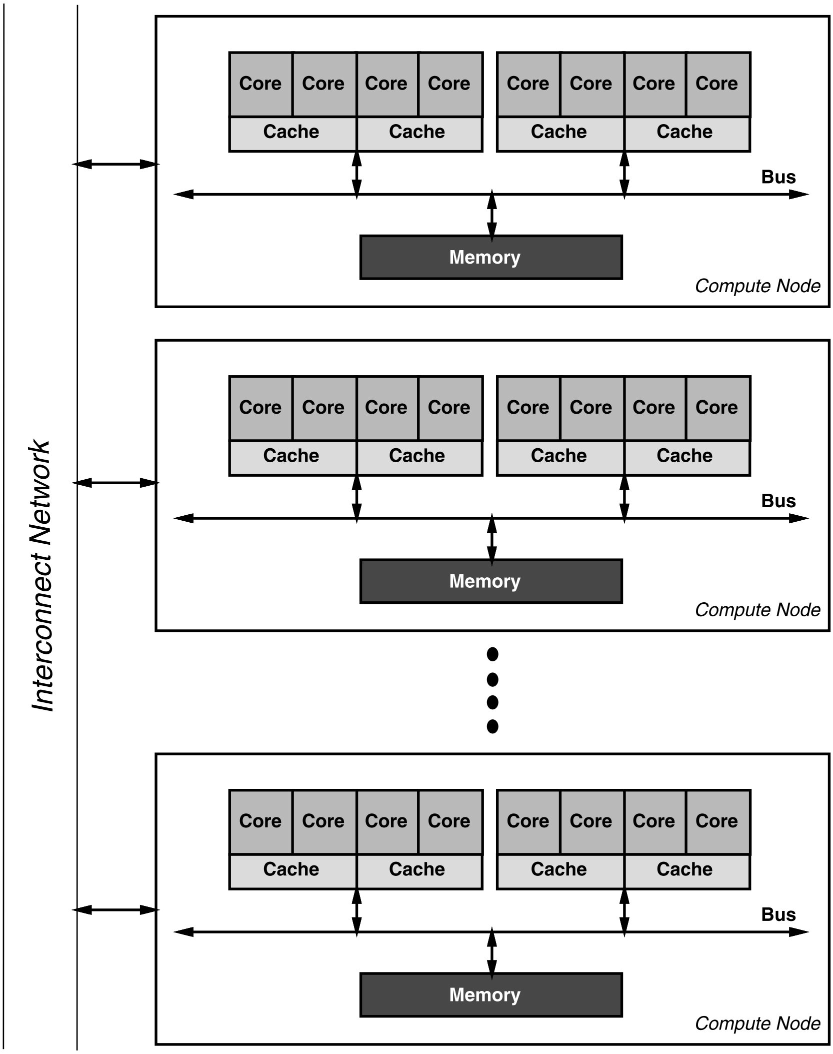


Figure 23.1 A generic schematic of a multicore-based cluster that consists of a number of compute nodes connected through some interconnect. Message passing is a form of internode communication. On each compute node, there are several processor chips that share the same memory and each chip has a number of processor cores that (partially) share cache.

hand, there is no explicit need for work load division. Communication and dynamic load balancing are either in an implicit form or carried out automatically, such as by using threads in OpenMP. If possible, some form of data locality associated with the threads will help with respect to performance.

Because of the wide availability of parallel systems, everyone will soon enough have to confront parallel computing. History, however, shows that writing parallel programs from scratch for distributed memory is hard and will probably remain so in future. Therefore, the keyword for good parallel software libraries is user friendliness, which first and foremost means hiding from the user low-level details about work division, communication, and load balancing. User friendliness also means a set of generic software components that can assist the user to easily carry out high-level parallelisation when necessary.

Let us consider the case of solving a single PDE by the finite element method. The two major computational tasks are the assembly and solution of linear systems. For this simple case, a parallel code can have the appearance of a serial one, provided that the underlying software libraries have the capability of at least (1) mesh partitioning, (2) concurrent discretisation on all the subdomains, and (3) solving a distributed linear system by the parallel implementation of an iterative method. Behind the scene, the involved libraries should adopt a distributed data structure, where one piece represents a subdomain and is assigned to one compute node, one processor, or even one core. An important requirement is that the libraries have flexible enough implementations to switch between fine-grained (one subdomain per core) and coarse-grained (one subdomain per compute node) parallelism. This is for tuning the performance on an arbitrary parallel system with a three-layer hierarchical architecture. For pre-multicore-era parallel software, it is therefore necessary to extend them to incorporate layered parallelism. We will return to this issue of flexibility at the end of the present section.

This simple case of solving a single PDE on a parallel computer is idealised because the user probably does not need to learn and program parallelism. Many real-life cases, however, require some effort of parallel programming on the part of the user. It can, for example, happen that a parallel algebraic preconditioner provided by some numerical library does not give fast enough convergence, so the user will want to implement a special parallel preconditioner. Another example is when a user wishes to reuse serial code in the parallelisation. This situation happens, for instance, when developing parallel code for multiphysics problems on the basis of serial single-physics software codes. To handle both these situations, we need to provide the user with some capability of 'high-level' parallel programming. Both MPI and/or OpenMP commands should be kept invisible because they involve too many details and another parallel programming tool can be used by the involved libraries. Instead, generic components for work division, data communication, and load balancing should be prepared as a library. In the case where the principle of additive Schwarz methods [68] is used to parallelise an existing serial PDE solver, a generic programming framework can also be implemented as a parallel library. This can be achieved by either class programming in C++ (see [70]) or programming generic functions in Python (see [69]).

23.5.3 Performance Issues

Since the compute nodes of a parallel system today do not share memory, some form of message passing will be used at least on the internode level to enable parallelism. Regarding programming tools, MPI is expected to continue its domination as the standard for message- passing programming. This is unlikely to change in the near future.

Many pre-multicore-era numerical libraries are already parallelised on the basis of subdomains and implemented using MPI calls. A straightforward use of these old parallel libraries on multicore-based systems is to start one MPI process on each available core, that is, in a flat MPI fashion. There are several potential performance disadvantages associated with this approach. First, when the number of available cores is large, the flat MPI approach results in fine-grained parallelism, that is, each subdomain can be very small. Besides the increased difficulty of mesh partitioning due to the increased number of subdomains, duplication between subdomain data structures (needed due to, e.g., ghost mesh elements and points) will also increase. This will lead to greater computational overhead, at least during finite element discretisation. Second, the total volume of message-based communication will increase with the number of subdomains. Third, since the communication speed can be quite different on the three architectural levels (internode, intranode, and on chip), an imbalanced distribution of communication overhead can arise to hamper the overall performance. This situation is typically not considered in standard mesh partitioning algorithms.

In addition, MPI's achievable performance can be questionable for fine-grained parallelism [67, 65]. Even if future MPI implementations automatically utilise appropriate mechanisms for sending and receiving messages on different architectural levels, flat MPI cannot be as good as a hybrid approach. In other words, MPI should perhaps be used only for internode communication, while some other form of (implicit) communication is used on the intranode and on-chip levels. OpenMP is a popular candidate for obtaining fine-grained parallelism on the latter two architectural levels. Recent studies, however, such as reported in [64, 66], suggest that an enhanced OpenMP standard (with, e.g., data locality capability) or a completely new programming tool for on-chip parallelism is needed.

Although we envision future parallel systems to have a unified conceptual design consisting of three architectural levels, the actual performance characteristic will vary from machine to machine. This depends on, for example, interconnect speed, processor speed, the number of nodes, processors, and cores. We therefore need future parallel software libraries to have the flexibility of switching between assigning one subdomain to one compute node, one processor, or one core. In other words, hybrid parallelism should be easily enabled by future parallel software. On our way to this goal, we should allow a layered software design with respect to work division (i.e., mesh partitioning in the context of solving PDEs), the choice of different parallel programming paradigms, and load balancing. Moreover, we should also provide generic software components and frameworks for the purpose of allowing users to easily implement their own "exotic" parallel codes with coarse-grained parallelism. Automatic code wrapping and a mixture of high-level and low-level programming languages can be important tools to use as well. Fine-grained parallelism is expected to be either delivered completely by libraries or enabled by using general-purpose programming models such as the MERGE model [71] for heterogeneous multicore systems.

To summarise, overall parallel performance will arise from efficient work division and MPI communication on the internode level, the appropriate choice of parallelism on the intranode and on-chip levels, plus possibly the efficiency of existing software provided by the user. Future parallel software libraries should have a layered and flexible design aiming at both performance and usability.

23.6 Automation and Code Generation

23.6.1 Automation

When scientific software reaches the maximum level of simplicity, we say that it *automates* the solution of the class of problems it can handle. Thus, one can say that MATLAB automates the solution of linear systems since the user interface is maximally simple and closely resembles the mathematical notation as demonstrated in table 23.1. We note that when we judge the level of simplicity, we must take into account the fact that mathematical notation is more flexible in terms of which symbols can be used than computer code, which is usually limited to (a subset of) the 128 characters in the ASCII character set.

Mathematical notation	Code
$x = A^{-1}b$	`x = A \ b`

Table 23.1 User input for solving a linear system $Ax = b$ in MATLAB.

We can also draw the conclusion that Diffpack and similar C++ libraries *do not* automate the solution of differential equations, since the user input required for any given differential equation does not closely correspond to the simplest specification of the equation, as demonstrated in table 23.2. This is not a surprise, since the solution of PDEs is much more difficult to automate than the solution of linear systems (which have a simple and well-defined mathematical structure).

```
for (int i = 1; i <= nbf; i++)
  for (int j = 1; j <= nbf; j++)
    for (int k = 1; k <= nsd; k++)
      elmat.A(i, j) += fe.dN(i, k) * fe.dN(j, k) * fe.detJxW();

for (int i = 1; i <= nbf; i++)
  elmat.b(i) += fe.N(i)*f(fe)*fe.detJxW();
```

Table 23.2 User input for discretising Poisson's equation $-\Delta u = f$ using the finite element method in Diffpack.

In the 1950s, it was important to learn how to compute $\sin(x)$ efficiently on a computer. Different codes were needed to handle different types of x arguments: single-precision float, double-precision float, single-precision complex, double-precision complex, or arrays with entries of the four types mentioned. Today, programmers just write `sin(x)` and do not worry about the underlying algorithm, the precision of `x`, or whether `x` is a scalar or an array[6]. The computation of the sine function is automated and the result comes with a guarantee of its accuracy.

Ultimately, scientific code writing as we know it today should and will be automated in the future. Application scientists should ideally only specify a mathematical problem, and possibly details about its solution, in a precise language close to what we use in scientific papers today. The creation of efficient computer code, adapted to the problem and the architecture, can be automated by specialised compilers, as we discuss below.

23.6.2 Efficiency vs. Generality

In general, we can say that for any automatic system which has a rich variation in its input space, there is a trade-off between efficiency and generality, as illustrated in figure 23.2. Thus, although MATLAB can be very efficient at solving some classes of linear systems, it cannot compete with highly tuned codes that can take advantage of domain-specific knowledge to solve linear systems with certain structures very efficiently. On the other hand, Diffpack can combine efficiency with generality for the solution of PDEs, but only since it does not automate the solution of PDEs but relies on problem-specific code for an essential part of the solution process.

[6] It is possible to write `sin(x)` for an array `x` in MATLAB and Python and in C++ if particular libraries are used, but not in Fortran and C.

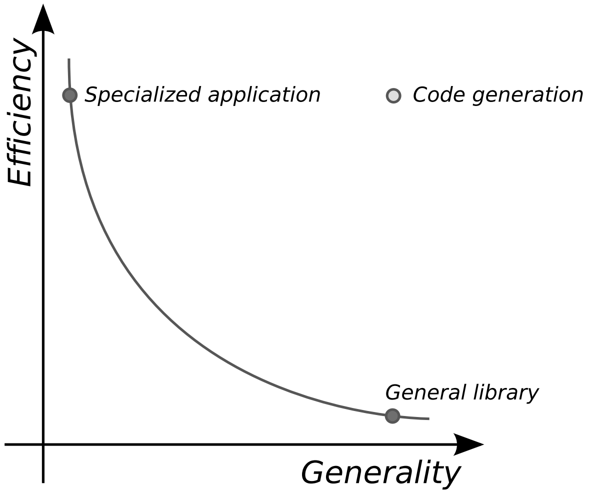


Figure 23.2 Trade-off between efficiency and generality for automated systems. Efficiency requires specialisation and general libraries cannot normally compete with specialised applications.

A grand challenge in the development scientific software thus lies in the design of automated systems that can combine efficiency with generality. We believe that the solution to this challenge lies in *code generation*.

23.6.3 Code Generation

With a system capable of generating highly efficient problem-specific code for any given problem from a certain class, one can potentially solve any problem from that class of problems with the same efficiency as the most specialised and optimised code, thus combining efficiency with generality.

To make this idea precise, consider an automated system that, for a certain class of problems, takes a set of input parameters specifying the problem in detail (user input) and from that specification computes the solution. Assume further that the user input can be partitioned into two sets, Input 1 and Input 2, such that an efficient solution of the problem requires specialised code that depends heavily on Input 1 but not so much on Input 2.

In the case of an automated system for the solution of PDEs, Input 1 can be the differential equation (and the finite element method used to discretise it), while Input 2 can be the mesh, boundary conditions, material data, coefficients, and other parameters. As illustrated in table 23.2, the computer code is highly specialised to the differential equation. On the other hand, one can implement highly efficient (but general) mesh libraries. [27, 47].

We now envision a just-in-time (JIT) compiler that from Input 1 can generate *executable code*. We further envision that the generated application-specific code can compute the solution to the original problem specified by both Input 1 and Input 2 given only Input 2. We illustrate this schematically in figure 23.3. As a result, we thus obtain an automated system that for given user input computes the solution to some problem specified by the input.

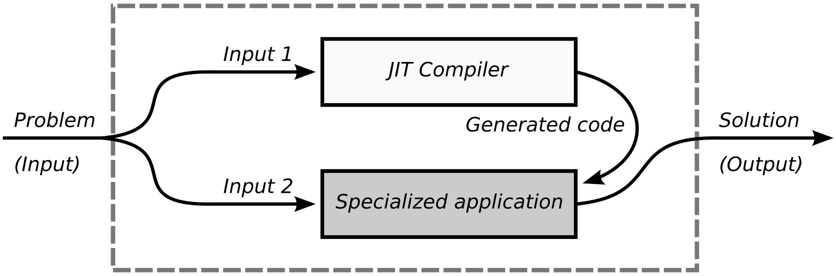


Figure 23.3 An automated system using a JIT compiler to generate a specialised application for a subset of its input.

The important point is that since the generated code that computes the solution is specialised to the given problem (in particular to Input 1), we are running a specialised application and can thus obtain efficiency. At the same time, if the JIT compiler can accept user input from a large class of problems, we obtain generality, thus combining efficiency with generality. User input is minimal, implying simplicity, and the code generator and associated libraries used to build the specialised code are reused from problem to problem, thereby contributing to reliability.

We emphasise that it is important to partition the input into two subsets Input 1 and Input 2, where Input 1 is minimal, in order to minimise the amount of code that needs to be generated. If we do so, we can to a large extent rely on general libraries and other reusable software components. As we have experienced first hand in our work on automated code generation, it is easier to develop and maintain computer code than it is to develop and maintain a code generator.

As demonstrated in [43, 42, 41, 40, 38], automatically generated code can sometimes significantly outperform handwritten optimised code (see also figure 23.4). This can seem counterintuitive but the code generator can automatically apply optimisations that are tedious or too complex to apply by hand. Such an optimising code generator (compiler) has been developed as part of the FEniCS project, which we discuss in the next section.

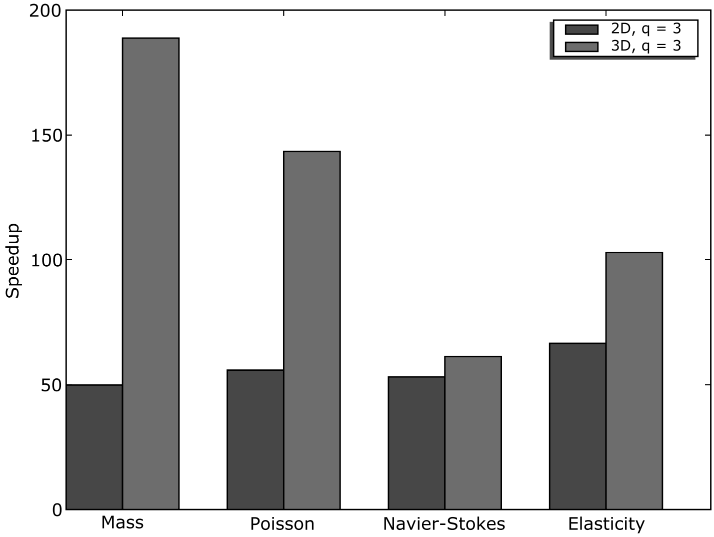


Figure 23.4 This figure shows the speedup for automatically generated C++ code for evaluating finite element variational forms, compared to standard handwritten code. The form compiler that generates the code makes use of special optimisation techniques that are difficult to implement manually [42].

23.7 FEniCS and the Automation of CMM

The FEniCS project is a collaborative free/open-source project towards the development of methodology and tools for the automation of computational mathematical modelling (CMM). [7, 39]. This includes building an automated system for the solution of PDEs. Below, we present an overview of the FEniCS project, including a short history, Simula's role, and the design of the FEniCS software.

23.7.1 History

The FEniCS project was initiated in 2003 at the University of Chicago. The initial conditions for the FEniCS project were DOLFIN [46] and FIAT [48, 49]. At the time, DOLFIN was a C++ library with tools for the solution of ordinary differential equations and PDEs, much like Diffpack. Here FIAT is a Python module for the generation of finite element basis functions and DOLFIN and FIAT remain two central software components in FEniCS today.

Originally DOLFIN was developed at the Chalmers University of Technology in 2002 as a replacement for Diffpack. At the time, Diffpack was used as the main computing platform at the Chalmers' Department of Computational Mathematics, but the proprietary nature of Diffpack meant that it could not be shown to students nor easily modified or improved. For this reason, a free/open-source replacement was needed. Since then, DOLFIN has undergone a large number of changes and has evolved from a monolithic C++ library sporting its own linear algebra and finite element backend to an automating system that outsources most of the computation to other FEniCS components and external libraries. In particular, DOLFIN relies on FIAT for the generation of finite element basis functions and on the form compilers FFC [45, 41, 40] and SyFi [1, 2] for JIT compilation of finite element variational forms and has a uniform object-oriented interface to any of a number of high-performance linear algebra backends, including PETSc [22], Epetra (Trilinos) [20], uBLAS [19], and MTL4 [13]. The C++ interface of DOLFIN has also been complemented with an easy-to-use MATLAB-like Python interface.

Since its start in 2003, the FEniCS project has grown considerably and now includes 12 official projects/components, with five new components on the horizon. The FEniCS project was developed in collaboration between a number of research institutions and universities, including (in order of appearance) the University of Chicago, Argonne National Laboratory, the Delft University of Technology, the Royal Institute of Technology (KTH), Simula Research Laboratory, Texas Tech University, and the University of Cambridge. Around 15 to 20 active developers spread out across these institutions are working constantly on improving the software. The FEniCS webpage[7] has around 10,000 unique visitors each month and the FEniCS software is downloaded around 1,000 times (total for all components) each month by users from more than 70 countries. The development of FEniCS is discussed openly on the FEniCS mailing lists, with 500 to 1000 monthly posts by developers and users.

23.7.2 The Role of Simula

In recent years, Simula Research Laboratory has become a main contributor to the FEniCS project. Simula is currently investing substantial resources in FEniCS, in terms of both research and development and testing and maintenance. This investment has had a very positive impact on the project.

Administrative tasks, such as maintenance of the FEniCS web server and mailing lists, testing on various platforms, building, and distribution, that were previously handled by a small group of researchers are now handled by professional scientific programmers. All FEniCS components are currently built and tested on three different platforms in six different configurations nightly to ensure reliability and cross-platform compatibility.

23.7.3 Design

Automation

The FEniCS project is based on the ideas presented in section 23.6.3. Thus, FEniCS partitions user input into two subsets: the differential equation (Input 1) and the computational domain, coefficients, parameters, and so forth (Input 2). The equa-

[7] www.fenics.org

tion is handed to a JIT compiler for finite element variational forms, which generates efficient low-level problem-specific C++ code that is then automatically compiled and executed (at runtime) for the given input data (Input 2). This is illustrated in figure 23.5, where we have also indicated which FEniCS components are being used.

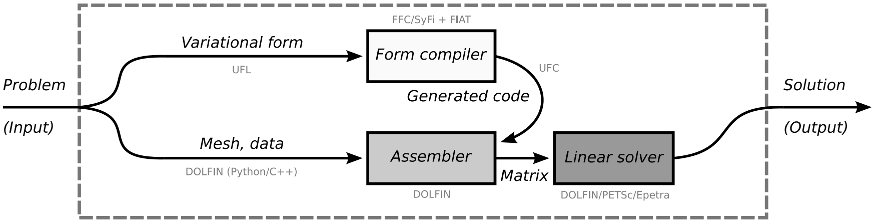


Figure 23.5 Here FEniCS automates the solution of finite element variational problems using a JIT compiler for finite element variational forms. For a given finite element variational form, the form compiler generates code for the computation of the so-called element matrix, which is used by an assembly algorithm to assemble the corresponding linear system on a given mesh. The linear system can subsequently be solved to obtain the solution of the given problem.

Currently FEniCS does not automate the solution of differential equations in the sense of section 23.6.1. Instead, FEniCS automates the solution of finite element variational problems. Thus, once a differential equation has been stated in the language of variational forms, it can be specified in FEniCS with maximal simplicity, as illustrated in table 23.3. For an extended discussion of automation and its application to CMM (see [44]).

Mathematical notation
$a(v,u) = \int_\Omega \operatorname{grad} v \cdot \operatorname{grad} u - \operatorname{div} v p + q \operatorname{div} u \, \mathrm{d}x$

Code
`a = (dot(grad(v), grad(u)) - div(v)*p + q*div(u))*dx`

Table 23.3 User input for specifying (part of) the variational problem for the Stokes problem or, equivalently, incompressible linear elasticity in FEniCS.

High-performance computing.

High-performance computing is supported in FEniCS based on a highly efficient implementation of computational mesh data structures [27] and wrappers for high-performance linear algebra libraries. We comment here on the linear algebra support in FEniCS which is implemented as part of DOLFIN.

Early versions of DOLFIN included data structures (sparse matrices and vectors) and algorithms (iterative linear solvers and preconditioners) implemented in C++ as part of DOLFIN. This implementation was later found to be insufficient and lacking in features and performance compared to available third-party libraries. At that point, the linear algebra in DOLFIN was replaced by PETSc [22], a well-known high-performance C library developed at Argonne National Laboratory. On top of PETSc, DOLFIN added a thin layer of wrappers to provide a simple and object-oriented interface to PETSc. Later, it was found desirable to also include wrappers for uBLAS [19], a simple template-based linear algebra library developed by the Boost project. The wrappers in DOLFIN were then extended to cover the use of both PETSc and uBLAS. Today, DOLFIN provides a uniform interface (based on polymorphism) to PETSc, Epetra (Trilinos), uBLAS, and MTL4, with preconditioners from Hypre [21] and ML [20]. This allows users to implement a single application and change the linear algebra backend with a simple switch.

Interfaces.

Two different user interfaces are provided by FEniCS: a C++ and a Python interface. The Python interface is generated semiautomatically by SWIG from the C++ interface. Thus, one can implement a FEniCS application in either C++ or Python, depending on taste. In both cases, the user is presented with a simple object-oriented interface which integrates well with external libraries.

Although FEniCS tries to automate as much as possible of the solution process, access is provided to all levels of the system, allowing users to handle special cases and using FEniCS components together with other libraries and special-purpose code in heterogeneous software environments. For example, one can at the potential cost of reliability edit the automated generated simulation code by hand by adapting it to special needs.

23.7.4 Software Map

All FEniCS components are developed and maintained by their respective authors. Although there is always a strong focus on compatibility and integration with other FEniCS components, there is also a strong focus on separation of concerns so that individual components can be singled out and used as part of other systems or applications. In particular, the Unified Form-assembly Code (UFC) interface specifies a fixed interface for code generation that allows the two form compilers FFC and SyFi to be used interchangeably to generate code. Figure 23.6 illustrates the relations between the various components of FEniCS.

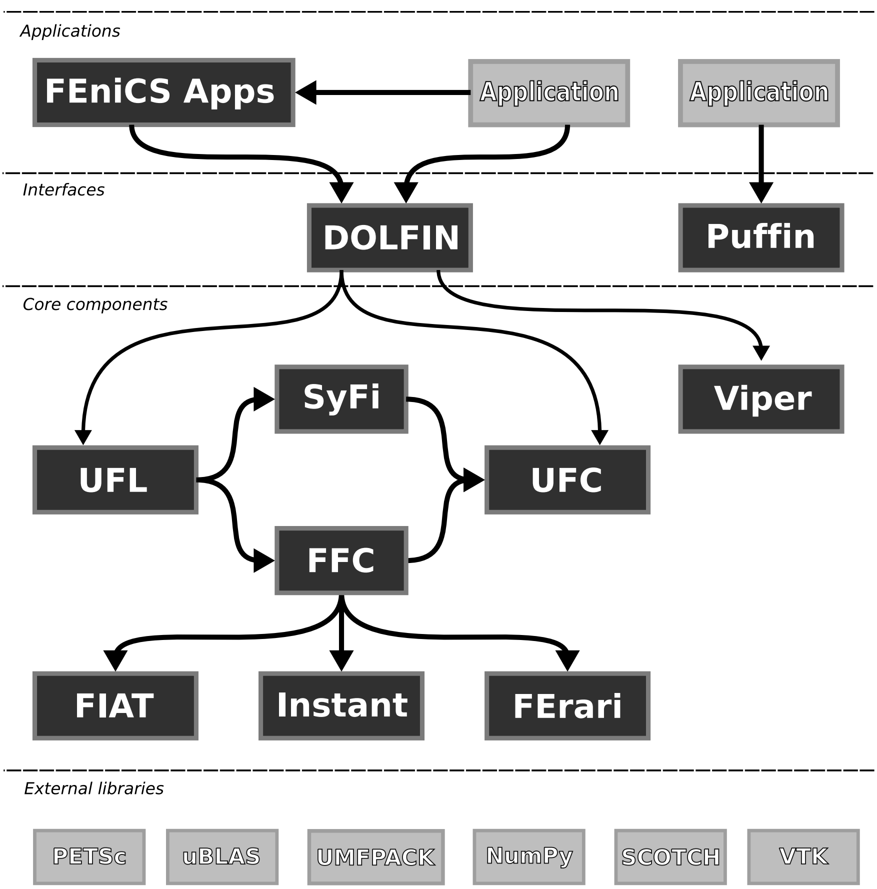


Figure 23.6 The FEniCS software map. A finite element application is either built with DOLFIN, C++, Python, or Puffin (using MATLAB). DOLFIN applies UFL to specify the problem in a notation close to mathematics. This specification is compiled, by SyFi or FFC, to a C++ code for computing element matrices and vectors, compliant with the UFC interface. This generated code is linked to DOLFIN to produce the application. Then FFC applies FIAT to compute finite element basis functions, Instant as a JIT compiler, and FErari to perform aggressive optimisation on the generated code.

23.8 Examples

We conclude this chapter by demonstrating how to use FEniCS to implement solvers for two different applications, first a very simple example and then an advanced example, both made easy by the automating framework provided by FEniCS.

23.8.1 Solving Poisson's Equation

The Poisson equation is a second-order PDE stating that the negative Laplacian $-\boldsymbol{\Delta} u$ of some unknown field $u = u(x)$ is equal to a given function f on a domain $\boldsymbol{\Omega}$, possibly amended by a set of boundary conditions for the solution u on some subset $\boldsymbol{\Gamma}$ of the boundary $\partial\boldsymbol{\Omega}$ of $\boldsymbol{\Omega}$:

$$\begin{aligned} -\boldsymbol{\Delta} u(x) &= f(x), \quad x \in \boldsymbol{\Omega}, \\ u(x) &= 0, \quad x \in \partial\boldsymbol{\Gamma}. \end{aligned} \tag{23.1}$$

Poisson's equation can be used as a simple model for gravity, electromagnetism, heat transfer, fluid flow, and many other physical processes. It also appears as the basic building block in a large number of more complex physical models, including the Navier-Stokes equations discussed below.

One would think that by this time, the question of how to solve Poisson's equation has been settled. Even today, more than 200 years after the birth of Siméon-Denis Poisson, researchers are still inventing new methods for solving the equation that he published in the *Bulletin de la Société Philomatique* in 1813 as a correction of an equation published earlier by Pierre-Simon Laplace.

Siméon Denis Poisson (1781-1840), inventor of Poisson's equation.

Poisson's equation is the canonical example that is always used to illustrate the benefits of a particular method or software.

To solve Poisson's equation with FEniCS, one must first rewrite equation (23.1) in *variational form*. This can formally be obtained by multiplying (23.1) with a *test function* v and then integrating by parts to obtain the following variational problem: Find $u \in V$ such that

$$\int_{\Omega} \operatorname{grad} v \cdot \operatorname{grad} u \, \mathrm{d}x = \int_{\Omega} v f \, \mathrm{d}x \tag{23.2}$$

for all test functions v in some suitable function space V. We now define a bilinear form a and a linear form L by

$$a(v,u) = \int_{\Omega} \operatorname{grad} v \cdot \operatorname{grad} u \, \mathrm{d}x, \tag{23.3}$$

$$L(v) = \int_{\Omega} v f \, \mathrm{d}x. \tag{23.4}$$

We can then state the variational problem for Poisson's equation in the following canonical form: Find $u \in V$ such that

$$a(v,u) = L(v) \tag{23.5}$$

for all $v \in V$.

FEniCS solves variational problems of the form (23.5) and lets the user specify the problem in close-to mathematical notation, thus automating the solution of variational problems in the sense of section 23.6.1. In table 23.4, we demonstrate how to implement a solver for Poisson's equation in Python using FEniCS.

```
from dolfin import *

# Create mesh and function space
mesh = UnitSquare(32, 32)
V = FunctionSpace(mesh, "CG", 1)

# Define variational problem
v = TestFunction(V)
u = TrialFunction(V)
f = Source(V)

a = dot(grad(v), grad(u))*dx
L = v*f*dx

# Define boundary condition
boundary = DirichletBoundary()
bc = DirichletBC(V, boundary, 0.0)

# Solve and plot solution
problem = VariationalProblem(a, L, bc)
u = problem.solve()
plot(u)
```

Table 23.4 Python implementation of a solver for Poisson's equation in FEniCS. We have left out the definition of the function f and the boundary Γ in the previous example.

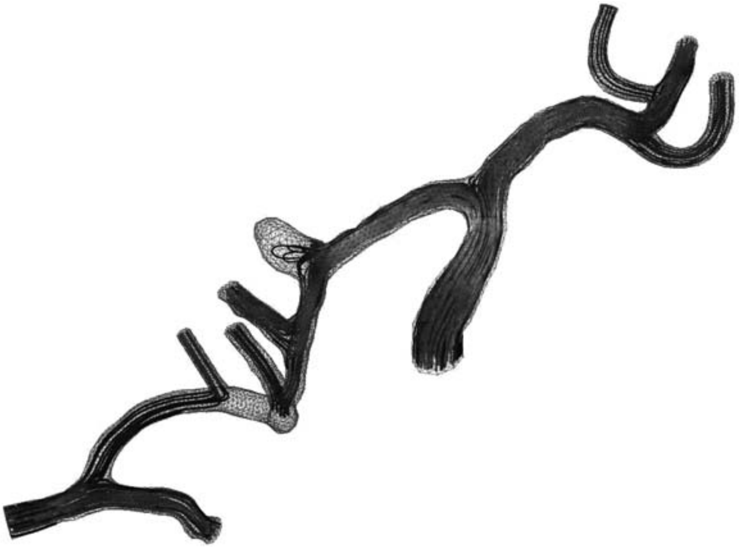


Figure 23.7 Simulation of blood flow through the circle of Willis with FEniCS (simulation by Kristian Valen-Sendstad).

The previous solver can easily be extended to general systems of advection-diffusion-reaction equations. Also FEniCS has been applied successfully to problems in computational finance (pricing of options), problems in electromagnetism (e.g., eigenspectrum analysis of waveguides), biomedical flow problems, including fluid-structure interaction, and simulation of cardiac electromechanics (including the solution of the bidomain equations). We now demonstrate how to use FEniCS for solving flow problems posed in the form of the incompressible Navier-Stokes equations.

23.8.2 Solving the Navier-Stokes Equations

FEniCS automates the solution of linear variational problems such as the one above for Poisson's equation. Although FEniCS provides support for solving nonlinear variational problems, more complex problems must still be programmed by the user. The high-level FEniCS framework, however, provides a number of powerful tools that allow a user to program solvers for complex problems with relative ease.

Consider, for example, the incompressible Navier-Stokes equations for the velocity u and pressure p of an incompressible fluid:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u - \nu \Delta u + \nabla p = f, \tag{23.6}$$

$$\nabla \cdot u = 0. \tag{23.7}$$

The Navier-Stokes equations are used as the basic model for the simulation of biomedical flow problems at Simula's Center for Biomedical Computing. Figure 23.7 shows the velocity field through a section of the circle of Willis, an arterial network at the base of the brain, obtained by solving the incompressible Navier-Stokes equations in FEniCS.

In table 23.5, we list a part of a simple Python implementation of a solver for the incompressible Navier-Stokes equations using FEniCS. As demonstrated, one can easily combine the basic tools provided by FEniCS—the specification of variational forms, the assembly of linear systems (the function `assemble()`), and the solution of linear systems (the function `solve()`)—to solve complex nonlinear problems like the incompressible Navier-Stokes equations. A corresponding implementation in Fortran or C++ would require many thousands of lines of code and would take months to implement. For comparison, we mention that when a user runs the Python script listed in table 23.5, FEniCS automatically generates 20,500 lines of highly efficient C++ code. Thus, a user can specify a problem or a solver with a minimal amount of high-level Python code, which translates into thousands of lines of low-level C++ code, which competes with or outperforms any handwritten Fortran or C++ code.

23.9 Conclusions

An important trend in scientific software (and in industrialised society in general) is automation. By grouping a series of complex operations into an automatic sequence, one can simplify common tasks such as the computation of `sin(x)`. Ongoing research is now making it possible to similarly create automatic systems for the solution of PDEs, based on the automatic generation of computer code and the integration of software written in Fortran, C, C++, and Python into heterogeneous software environments.

Even if FEniCS can be said to represent the current state of the art in scientific programming, we still need to write large amounts of code and build complex systems. In some respects, FEniCS is a complex beast with intricate dependencies and elaborate schemes for code generation. One of the keys to reaching the next level will be to further extend our current work on componentisation and standardisation.

A successful example of this strategy is the UFC interface for finite element code generation mentioned in section 23.7.4. Version 1.0 of UFC was released in June 2007 and the interface has remained stable and virtually unchanged (with only minor additions) since then. In contrast, the problem-solving environment DOLFIN undergoes large changes (improvements), almost daily, with frequent updates (improvements) of its interface. The standardisation on UFC as a fixed interface for code generation, however, has greatly simplified the work on both DOLFIN and the form compilers FFC and SyFi. Through further standardisation, one can build fixed components that can be used as building blocks without needing to worry about their inner workings, much in the same way that we rarely worry about the inner workings of GCC (although it can happen at occasions).

```
# Tentative velocity step
a0 = dot(v, u)*dx + k*nu*dot(grad(v), grad(u))*dx
L0 = dot(v, u0)*dx + k*dot(v, f)*dx - \
     k*dot(v, mult(grad(u0), u0))*dx

# Poisson problem for the pressure
a1 = dot(grad(q), grad(p))*dx
L1 = -(1.0/k)*q*div(us)*dx

# Velocity update
a2 = dot(v, u)*dx
L2 = dot(v, us)*dx - k*dot(v, grad(p1))*dx

# Assemble matrices
A0 = assemble(a0)
A1 = assemble(a1)
A2 = assemble(a2)

# Time loop
t = 0.0
while t < T:

    # Compute tentative velocity
    b = assemble(L0)
    [bc.apply(A0, b) for bc in problem.bcv]
    solve(A0, us.vector(), b, gmres, ilu)

    # Compute p1
    b = assemble(L1)
    [bc.apply(A1, b) for bc in problem.bcp]
    solve(A1, p1.vector(), b, gmres, amg_hypre)

    # Compute u1
    b = assemble(L2)
    [bc.apply(A2, b) for bc in problem.bcv]
    solve(A2, u1.vector(), b, gmres, ilu)

    # Propagate values to next time step
    t += dt
    u0.assign(u1)

    # Plot solution
    plot(u1)
    plot(p1)
```

Table 23.5 Python implementation of a solver for the incompressible Navier-Stokes equations in FEniCS, using a simple Chorin pressure-correction scheme.

Thus, whatever ideas drive the development of the next generation software for PDEs, that software can hopefully use one or more FEniCS components as its basic building blocks.

Acknowledgements

Many people have contributed to developing the software described in this chapter. The authors would like to acknowledge some very important individuals.

Are Magnus Bruaset played a key role in the development, maintenance, and distribution of Diffpack.

Kent-Andre Mardal contributed with several modules in Diffpack and later with the SyFi tool in FEniCS.

Ola Skavhaug has made significant contributions to the FEniCS project by developing the FEniCS plotting library Viper and by building and supporting a software infrastructure for FEniCS at Simula.

Garth N. Wells (University of Cambridge) works together with Anders Logg as one of the two maintainers of DOLFIN and is responsible for a significant portion of DOLFIN's design.

Martin Sandve Alnæs is one of the main architects behind the UFC interface and is currently leading a new project towards the creation of a common interface for the expression of variational forms (UFL).

We also mention the following contributors to the FEniCS project: Johan Hake, Johan Hoffman, Johan Jansson, Claes Johnson, Dmitry Karpeev, Benjamin Kehlet, Robert C. Kirby, Matthew G. Knepley, Kristian Oelgaard, Johannes Ring, Marie Rognes, L. Ridgway Scott, Andy R. Terrel, Åsmund Ødegård, Magnus Vikstrøm, and Ilmar Wilbers.

References

[1] M. S. Alnæs and K.-A. Mardal. *SyFi*, 2009. http://www.fenics.org/wiki/SyFi/.

[2] M. S. Alnæs and K.-A. Mardal. Symbolic computations and code generation for finite element methods, 2009.

[3] O.-J. Dahl and K. Nygaard. Simula–a language for programming and description of discrete event systems. introduction and user's manual. *NCC Publ.*, (11), 1965.

[4] M. S. Alnæs and K.-A. Mardal. Symbolic computations and code generation for finite element methods. *journal*, 2008.

[5] Numerical Python software package. http://sourceforge.net/projects/numpy.

[6] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[7] FEniCS software collection. http://www.fenics.org.

[8] J. H. Ring, H. P. Langtangen, R. E. Bredesen, and I. Wilbers. SciTools software package. http://code.google.com/p/scitools.

[9] SWIG software package. http://www.swig.org.

[10] P. Peterson. F2PY software package. http://cens.ioc.ee/projects/f2py2e.

[11] Diffpack software package. http://www.diffpack.com.

[12] H. P. Langtangen. The FEMDEQS program system. Research report in mechanics, Mechanics Division, Department of Mathematics, University of Oslo, 1989.

[13] J. Siek and A. Lumsdaine. A modern framework for portable high-performance numerical linear algebra. *Advances in Software Tools for Scientific Computing*. Springer, 1999.

[14] T. L. Veldhuizen. Blitz++: The library that thinks it is a compiler. *Advances in Software Tools for Scientific Computing*. Springer, 1999.

[15] T. L. Veldhuizen and M. E. Jernigan. Will C++ be faster than Fortran? *Scientific Computing in Object-Oriented Parallel Environments*, Lecture Notes in Computer Science, pages 49–56. Springer, 1997.

[16] C. Pflaum and Z. Rahimi. Automatic parallelization of staggered grid codes with expression templates. *International Journal of Computational Science and Engineering*, 2009.

[17] E. Arge, A. M. Bruaset, P. B. Calvin, J. F. Kanney, H. P. Langtangen, and C. T. Miller. On the efficiency of C++ in scientific computing. *Mathematical Models and Software Tools in Industrial Mathematics*, pages 91–118. Birkhäuser, 1997.

[18] U. T. Mello and I. Khabibrakhmanov. On the reusability and numeric efficiency of C++ packages in scientific computing. *Proceedings of the ClusterWorld Conference and Expo*, 2003.

[19] uBLAS software package. http://www.boost.org/libs/numeric/ublas/doc/.

[20] Trilinos software package. http://trilinos.sandia.gov/.

[21] Hypre software package. http://www.llnl.gov/CASC/hypre/.

[22] PETSc software package. http://www.anl.gov/petsc.

[23] Netlib repository of numerical software. http://www.netlib.org.

[24] X. Cai, H. P. Langtangen, and H. Moe. On the performance of the Python programming language for serial and parallel scientific computations. *Scientific Programming*, 13(1):31–56, 2005.

[25] H. P. Langtangen and X. Cai. On the efficiency of Python for high-performance computing: A case study involving stencil updates for partial differential equations. *Modeling, Simulation and Optimization of Complex Processes*, pages 337–358. Springer, 2008.

[26] M. S. Alnæs, A. Logg, K.-A. Mardal, O. Skavhaug, and H. P. Langtangen. Unified framework for finite element assembly. *International Journal of Computational Science and Engineering*, 2009.

[27] A. Logg. Efficient representation of computational meshes. *International Journal of Computational Science and Engineering*, 2009.

[28] X. Cai and H. P. Langtangen. Parallelizing PDE solvers using the Python programming language. *Numerical Solution of Partial Differential Equations on Parallel Computers*, volume 51 of *Lecture Notes in Computational Science and Engineering*, pages 295–325. Springer, 2006.

[29] H. P. Langtangen. A case study in high-performance mixed-language programming. *Applied Parallel Computing – State of the Art in Scientific Computing*, Lecture Notes in Computer Science, pages 36–49. Springer, 2007.

[30] C. C. Douglas and H. P. Langtangen. General methods for implementing reliable and correct software: C, C++ and Python. *Accuracy and Reliability in Scientific Computing*. SIAM, 2005.

[31] B. Einarsson. General methods for implementing reliable and correct software: Fortran. *Accuracy and Reliability in Scientific Computing*. SIAM, 2005.

[32] J. J. Barton and L. R. Nackman. *Scientific and Engineering C++ – An Introduction with Advanced Techniques and Examples*. Addison-Wesley, 1994.

[33] Y. Shapira. *Solving PDEs in C++: Numerical Methods in a Unified Object-Oriented Approach*. Computational Science and Engineering. SIAM, 2006.

[34] M. Metcalf. *Effective Fortran 77*. Oxford University Press, 1985.

[35] M. Metcalf, J. K. Reid, and M. Cohen. *Fortran 95/2003 Explained*. Oxford University Press, 2004.

[36] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[37] H. P. Langtangen. *Computational Partial Differential Equations - Numerical Methods and Diffpack Programming*. Texts in Computational Science and Engineering, vol 1. Springer, 2nd edition, 2003. 855 p. http://www.diffpack.com.

[38] R. C. Kirby and A. Logg. Benchmarking domain-specific compiler optimizations for variational forms. *To appear in ACM Transactions on Mathematical Software*, (12), 2008.

[39] A. Logg. Automating the finite element method. *Arch. Comput. Methods Eng.*, 14(11):93–138, 2007.

[40] R. C. Kirby and A. Logg. Efficient compilation of a class of variational forms. *ACM Transactions on Mathematical Software*, 33(10), 2007.

[41] R. C. Kirby and A. Logg. A compiler for variational forms. *ACM Transactions on Mathematical Software*, 32(9):417–444, 2006.

[42] R. C. Kirby, A. Logg, L. R. Scott, and A. R. Terrel. Topological optimization of the evaluation of finite element matrices. *SIAM J. Sci. Comput.*, 28(8):224–240, 2006.

[43] R. C. Kirby, M. G. Knepley, A. Logg, and L. R. Scott. Optimizing the evaluation of finite element matrices. *SIAM J. Sci. Comput.*, 27(6):741–758, 2005.

[44] A. Logg. *Automation of Computational Mathematical Modeling*. PhD thesis, Chalmers University of Technology, Sweden, 2004.

[45] A. Logg et al. FFC. URL: http://www.fenics.org/ffc/.

[46] A. Logg, G. N. Wells, et al. DOLFIN. URL: http://www.fenics.org/dolfin/.

[47] M. G. Knepley and D. A. Karpeev. Mesh algorithms for PDE with Sieve I: Mesh distribution. Technical Report ANL/MCS-P1455-0907, Argonne National Laboratory, February 2007. Submitted to Scientific Programming.

[48] R. C. Kirby. FIAT: A new paradigm for computing finite element basis functions. *ACM Trans. Math. Software*, 30:502–516, 2004.

[49] R. C. Kirby. Optimizing FIAT with Level 3 BLAS. *ACM Transactions on Mathematical Software*, 32(2):223–235, June 2006.

[50] X. Cai, E. Acklam, H. P. Langtangen, and A. Tveito. Parallel computing. *Advanced Topics in Computational Partial Differential Equations – Numerical Methods and Diffpack Programming*, Lecture Notes in Computational Science and Engineering, pages 1–56. Springer, 2003.

[51] A. M. Bruaset, X. Cai, H. P. Langtangen, and A. Tveito. Numerical solution of PDEs on parallel computers utilizing sequential simulators. *Scientific Computing in Object-Oriented Parallel Environments*, Lecture Notes in Computer Science, pages 161–168. Springer, 1997.

[52] X. Cai. Domain decomposition. *Advanced Topics in Computational Partial Differential Equations – Numerical Methods and Diffpack Programming*. Springer, 2003.

[53] A. M. Bruaset, H. P. Langtangen, and G. W. Zumbusch. Domain decomposition and multilevel methods in Diffpack. *Proceedings of the 9th Conference on Domain Decomposition*. Wiley, 1997.

[54] K.-A. Mardal, G. W. Zumbusch, and H. P. Langtangen. Software tools for multigrid methods. *Advanced Topics in Computational Partial Differential Equations – Numerical Methods and Diffpack Programming*, Lecture Notes in Computational Science and Engineering, pages 97–152. Springer, 2003.

[55] K.-A. Mardal and H. P. Langtangen. Mixed finite elements. *Advanced Topics in Computational Partial Differential Equations – Numerical Methods and Diffpack Programming*, Lecture Notes in Computational Science and Engineering, pages 153–198. Springer, 2003.

[56] H. P. Langtangen and H. Osnes. Stochastic partial differential equations. *Advanced Topics in Computational Partial Differential Equations – Numerical Methods and Diffpack Programming*, Lecture Notes in Computational Science and Engineering, pages 257–320. Springer, 2003.

[57] G. Wilson. The history of the development of parallel computing, 1994.

[58] M. Wolfe. Compilers and more: Parallel programming made easy?, 2008.

[59] Top500 supercomputing sites. http://www.top500.org, November 2008.

[60] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI – Portable Parallel Programming with the Message-Passing Interface*. The MIT Press, 2nd edition, 1999.

[61] B. Chapman, G. Jost, and R. van der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*. The MIT Press, 2007.

[62] J. Shalf. The new landscape of parallel computer architecture.*Journal of Physics: Conference Series*, 78:012066, 2007.

[63] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick. The landscape of parallel computing research: A view from Berkeley. Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, 2006.

[64] A. Buttari, J. Dongarra, J. Kurzak, J. Langou, P. Luszczek, and S. Tomov. The impact of multicore on math software. *Proceedings of PARA 2006*, volume 4699 of *Lecture Notes in Computer Science*, pages 1–10. Springer Verlag, 2007.

[65] J. Psota and A. Agarwal. rMPI: message passing on multicore processors with on-chip connection. *Proceedings of HiPEAC 2008*, volume 4917 of *Lecture Notes in Computer Science*, pages 22–37. Springer Verlag, 2008.

[66] M. Nordén, H. Löf, J. Rantakokko, and S. Holmgren. Dynamic data migration for structured AMR solvers. *International Journal of Parallel Programming*, 35(5):477–491, 2007.

[67] L. Chai, Q. Gao, and D. K. Panda. Understanding the impact of multi-core architecture in cluster computing: A case study with Intel dual-core system. *Proceedings of Seventh IEEE International Symposium on Cluster Computing and the Grid.* 2007.

[68] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[69] J. K. Nilsen, X. Cai, B. Høyland, and H. P. Langtangen. Simplifying parallelization of scientific codes by a function-centric approach in Python. In preparation, 2008.

[70] X. Cai and H. P. Langtangen. Developing parallel object-oriented simulation codes in Diffpack. *Proceedings of the Fifth World Congress on Computational Mechanics*, Vienna University of Technology, 2002.

[71] M. D. Linderman, J. D. Collins, H. Wang, and T. H. Meng. Merge: a programming model for heterogeneous multi-core systems. *Proceedings of ASPLOS'08*, pages 287–296. ACM, 2008.

24

SOFTWARE ENGINEERING — WHY, WHAT, HOW AND WHAT'S NEXT

Dag I. K. Sjøberg and Stein Grimstad

Why Software Engineering Research is Important

Software systems form the foundation of the modern information society. As such, whether stand-alone or embedded in other technological systems, they are crucial to the continuing development of global society in all its spheres: social, political, economic, and cultural. For example, solving large-scale humanitarian and environmental problems calls for sophisticated software systems of the highest quality.

Nevertheless, most of today's software development projects face serious difficulties in reaching the expected level of quality, as well as in meeting schedules and financial budgets[1]. Billions of dollars are squandered annually on poorly planned and inefficiently executed software development projects. We all have seen software scandals reported in the media.

The sociotechnical nature of software development sets it apart from other engineering disciplines: The cost of a software system arises from its highly complex human- and technology-intensive development process. Many such systems are among the most complex things ever created by man. Future demands will add further to this complexity.

Dag I. K. Sjøberg · Stein Grimstad
Simula Research Laboratory

Dag I. K. Sjøberg
Department of Informatics, University of Oslo, Norway

[1] See, for example, Inside Risks comments in Comm. of the ACM, www.csl.sri.com/users/neumann/insiderisks.html.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_24,

The Software Engineering department of Simula addresses these challenges; that is, our motivation for the research conducted in the department is directed at enabling more high-quality, efficient, and predictable software development for industry, commerce, and services.

How Software Engineering Research is Conducted at Simula

In August 1999, the "Industrial Systems Development" research group was created in the Department of Informatics, University of Oslo, with a focus on empirical software engineering. This group became the core of Simula's SE department in 2001. By developing a research group almost from scratch, and combining that with the resources available at Simula at that time plus strong management and focused research, we were offered a unique opportunity. The research approach of our department was centred around the following aspects: industry relevance and transfer, variety of research methods, support environments, scientific databases, and research management.

Software engineering is about developing, maintaining, and managing high-quality software systems in a cost-effective and predictable way. Software engineering research studies the real-world phenomena of software engineering and concerns the development of new, or modification of, existing, technologies (process models, methods, techniques, tools, or languages) to support software engineering activities; and the evaluation and comparison of the effect of using such technology in the often highly complex interaction of individuals, teams, projects, and organisations, along with various types of tasks and software system. Science disciplines that study real-world phenomena, that is, the empirical sciences, of necessity use empirical methods, namely, the gathering of information based on systematic observation and experiment, rather than the use of deductive logic or mathematics. The major scientific challenge our department presently focuses on, within its specific areas (effort estimation, testing, model-driven development, maintenance, process improvement), is in *quantifying* and *understanding* the effects of using various process models, methods, techniques, and tools in various industrial situations. In other words, we aim to provide cost-benefit analyses over variation in software developers, teams, projects, and organisations, and across various types of activities and software systems. In the areas for which we have a sufficient understanding of such effects, we also propose new or modified technologies (e.g., in the area of software effort estimation and software testing tools).

The ultimate goal of software engineering research is to support practical software development. In our department, we have therefore posed the question: How do we convince practitioners and managers in industry that the results of our empirical studies are relevant to them? Relevance may be divided into the investigated topic and the *implications of the results*. Preferably, the topics we study will have their roots in problems or challenges experienced by the software development industry itself. Nevertheless, the history of science includes many examples of studies in which the scope of the study's relevance was only first understood by other researchers many years after the research had been conducted. As an illustration, essential features of general purpose, object-oriented languages today are based on research that aimed at the development of a special-purpose programming language for simulating discrete event systems: Simula. Moreover, our department attempts to conduct empirical studies that are as realistic as is practically possible. Consequently, we have involved several hundred companies in our studies since the founding of Simula. The emphasis on industrial collaboration is reflected in our department's motto: "The industry is our lab". Strong links to industry are also necessary for the direct technology transfer of our research results. Our results are also packaged into a form that is suitable for being taught at university and industry courses, seminars, and so forth, as well as for articles published in newspapers, in magazines, and on the Internet.

Combining practical relevance with scientific rigour, as well as acquiring knowledge about a wide spectrum of aspects of software development, requires that appropriate research methods be applied. We therefore do not belong to any one particular "school" with respect to research method, but rather use a variety of methods: controlled experiments, (multiple) case studies, action research, systematic review, (personal opinion) surveys, and simulations.

For the practical conduct and analysis of our studies, our department develops a variety of supporting tools, from simple script-based tools to sophisticated, commercial-quality environments. For example, the Simula Experiment Support Environment (SESE) has continuously been developed since the start of Simula and supports the logistics and accurate data collection of large-scale, industrial experiments. To improve the qualitative data obtained from software engineering experiments, our department developed both a method and a corresponding support tool for collecting feedback from subjects during experiments.

Part of our strategy is to construct databases of primary studies related to specific topic areas and use them to address specific research questions. These databases are also shared with external research groups, partly through special agreements.

The SE department fully exploits the unique opportunity at Simula to use its resources in the way the department itself finds optimal (the same person manages research, budget, and administration). For example, the "Simula effect" has been partly achieved by spending a relatively high proportion of the budget on external consultants, both for support tasks and participation in experiments and other studies, and on infrastructure and apparatus to conduct studies. This is mainly achieved at the expense of employing a larger number of researchers.

What has been Achieved

The goal of being able to quantify the effect of any software engineering technology can hardly ever be completely achieved, but Simula has made a significant step in pursuing this goal. Our department has advanced the state of the art regarding realism and scale of empirical studies in software engineering. In fact, the number and extent of *controlled experiments* and other studies involving industrial participants (as opposed to students) have become a trademark of our department. By January 2009, that is, in Simula's first eight years, 262 companies (including public sector agencies) from 24 countries, consisting of 2,730 professionals, have taken part in 37 experiments. About 1,100 of the participants were consultants hired by Simula. Most of the other participants were either participating in seminars held by Simula, which included an experiment, or the participants took part in experiments as part of an agreement that Simula offer a seminar. In addition to the most comprehensive experiments conducted to date, with respect to the number of professionals taking part (see chapter 29), our department has also conducted the first real-life experiment on bidding processes.

Case studies are crucial in order to study software engineering phenomena in complex, real-life environments. Our department has conducted ten case studies as part of software process improvement projects funded by the Research Council of Norway, including one of the few real-life studies on the effect of using the Unified Modelling Language (UML). Our department took a novel approach and imposed more control on (multiple) case studies: In one study on variability and reproducibility in software engineering, four companies were hired to develop the same (functionally equivalent) system independently of each other.

In an investigation reported in an article published in *Information and Software Technology* [1], Simula was found to be the leading research group on *systematic reviews* in software engineering. As part of the work on systematic reviews, Simula has built two scientific databases of, respectively, software development effort-estimation research and controlled experiments in software engineering. Both databases have been frequently used by other research groups.

A goal of the empirical studies of the research community is to acquire a better understanding of the individual, organisational, and technological processes involved in software engineering. In compliance with classical research disciplines, such understanding ideally should be formulated in terms of *theories*. Our department has carried out a systematic review revealing there has been little use of theories that articulate software engineering phenomena. Some initial guidelines for how to develop theories in the field have been proposed.

In addition to methodological innovations, our department has made significant contributions to the state of research in many aspects of the software development life cycle, such as requirements specification, software effort estimation, model-driven development, and software maintenance. For example, our department hosts the premier research group on judgment-based effort estimation. Among its contributions are improved terminology, processes, guidelines, and principles of judgment-based effort estimation. A debate between Magne Jørgensen and Barry Boehm (the founder of the COCOMO estimation model) on the strengths and weaknesses of model-based and judgment-based effort estimation is published in *IEEE Software* [2]; see more information about this work in chapter 26.

A number of external indicators that speak to the quality of our research output are a matter of record: Our department is number three among 1,361 institutions worldwide, in a ranking published in *Journal of Systems and Software* [3]. The ranking is the twelfth in an annual series and is based on frequency of publications in the leading software engineering journals. Both the previous and present Minister of Research and Education of Norway have publicly referred to this ranking, and a previous version of the ranking was mentioned in Report No. 17 (2006–2007) to the Norwegian Parliament on Information and Communication Technologies. Further, in an article in the *Communications of the ACM*, Simula is, together with the Fraunhofer Center (Germany and Maryland, USA) and the National Information and Computing Technology Centre of Australia, described as an institution that is ”clearly focused on the blending of research/theory and practice” [4]. The publications of our department are well-cited and departmental researchers are invited to give keynote addresses and write articles on the research conducted at Simula. Magne Jørgensen, who was number one among 3,918 researchers in the ranking reported in *Journal of Systems and Software* [3], has for the past two years written a regular column in *Computerworld Norway*. Finally, our department regularly organises seminars for industry, which attract 60–120 participants.

What's Next

Our motto is even more relevant and fitting today than it was when we started in 2001: The industry is our lab! The SE department's main goal is, and will continue to be, to provide novel, practical, and cost-effective solutions for important, enduring, and challenging problems faced by the IT industry. Our department will continue focusing on technological innovation combined with carefully planned and conducted empirical studies in order to achieve our ambitious goal. Moreover, because software engineering is typically performed by humans in organisations, our department will continue its collaborations with researchers from other disciplines, such as psychology, education, economics, statistics, and management.

Our persistent focus on industrially relevant research topics is starting to reap rewards. We are now in a position in which we can be more ambitious regarding industrial collaborations. For example, we have been working on establishing a large-scale, strategic research alliance with Det Norske Veritas. The purpose of the joint research and development programme is to deliver novel, model-based software engineering approaches to the development, verification, operation, and maintenance of software-intensive control and monitoring systems in the maritime and energy sectors. An agreement has now been reached between Simula and DNV, where DNV offers substantial funding for two years.

The prioritisations of financial and human resources, industrial collaborations, and the applications for research grants will be guided by the potential for their impact on software development productivity and quality. This means, amongst other things, that every project in the SE department is required to demonstrate ambitious goals related to our desire to make a difference in the software industry. Although we are proud of our achievements thus far, we expect to grow considerably in the future, both in terms of human resources and in research achievements that will make a significant and positive impact on the landscape of industry.

References

[1] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15, 2009.

[2] M. Jørgensen and B. Boehm. Software development effort estimation: Formal models or expert judgment? *IEEE Software*, (Jan/Feb), 2009.

[3] W. E. Wong, T. Tse, R. L. Glass, V. R. Basili, and T. Chen. An assessment of systems and software engineering scholars and institutions (2001–2005). *Journal of Systems and Software*, 81(6):1059–1062, 2008.

[4] R. L. Glass. A deja-vu look at software engineering researchers who care about practice. *Communications of the ACM*, 50(8):21–23, 2007.

25

A MATTER OF JUDGEMENT

An interview with Magne Jørgensen by Dana Mackenzie

One of the few constants in the world of software engineering is that projects always take longer to finish than you expect. The classic example is a video game called "Duke Nukem Forever," which was announced in 1997 by its manufacturer, 3D Realms... and still had not been released as of early 2009. The joke among video game fans is that the initials "DNF" stand for "Did Not Finish."

Although "Duke Nukem" is an extreme case, software development projects on the average take 30 to 40 per cent longer to complete than their developers estimate. And this figure has not shown any tendency to decrease over the years. Not only are software developers over-optimistic about the time needed to complete their projects, they are also over-optimistic about their own estimation abilities. When asked for a *range* of completion times, intended to capture the true development time 90 per cent of the time, project managers typically provide a too-narrow window that has a "hit rate" of only 60 to 70 per cent.

According to Magne Jørgensen, most research on effort estimation in the software engineering industry has been directed at "model-based estimation." The most widely known example is COCOMO (short for Constructive Cost Model), which was developed by Barry Boehm, a mathematician, programmer and manager at TRW Aerospace in 1981, and significantly updated in 1997. The most basic version of COCOMO is a simple set of equations that convert the number of lines of code (a basic estimate of project size) into an estimate of person-months required to complete the project. These are empirical equations that were based on a number of actual projects of various types, sizes, languages and platforms. The 1997 update was intended to apply to more modern styles of software development, and it also included measures of project size, such as "function points" (introduced by Alan Albrecht of IBM in 1979 and popularized in the mid-1980s), that were believed to be more meaningful than lines of code.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_25,

In spite of the effort that has gone into making models such as COCOMO more predictive, Jørgensen says, the fact of the matter is that they have not improved over time. The most telling evidence, he says, is that most managers do not trust them. The *de facto* standard industry practice is still "judgement-based estimation," in which software engineers base their estimates of completion time on some poorly-understood combination of gut feelings, past experience, negotiation among the different players in the project, and possibly inaccurate or irrelevant information provided by the client.

Although this estimation method may not even sound like a method at all, Jørgensen believes that it is worth taking seriously. In fact, he argues, managers may be better off using their judgement, if they take certain easy steps to make it more accurate. In the interview below, he explains why trusting the experts isn't such a bad idea; how to improve expert judgement; and why it might be a good idea for project managers to pay a little bit more attention to psychologists (and a little bit less attention to consultants with fancy equations).

"Magne, how did you get started in software engineering, and in particular this area of effort estimation?"

"My PhD research had an element of psychology in it. I was looking at what makes code complex and difficult to maintain. Most software engineering research is very technical, but it's obvious that it's actually people that are actually doing this work. It's a paradox, in a way, that our education is not more oriented toward people. We need to answer questions like these: What makes things complex? What makes people make errors? And the question I ended up with, what makes people overoptimistic?

"As a side product of my PhD thesis, which was about software maintenance, I had a lot of data, and if you have a lot of data you can make an estimation" model. That was the beginning of my research on effort estimation."

"So at that point, were you doing model-based estimation?"

"Yes."

"How did you start realizing that it was not working very well, and that you needed to look at judgement-based estimation?"

"That is a little bit strange. Currently, most estimates are based on expert judgement, as they have been since the 1970s. It is also evident that most estimates are overoptimistic and somehow very wrong, very unrealistic. So as in most other software engineering fields, we think we need to have a model that replaces the person and the need for expert judgement. We need something objective. That is where 95 per cent of the estimation research has been, and it has been near 100 per cent in the past. But even after all these years with estimation models, we are not even close to building an estimation model that is good. Most estimation models are in most contexts very poor.

Magne Jørgensen

"It seemed to me like a good idea to change the approach. If something doesn't work, maybe we shouldn't spend another forty years on even more advanced models. The alternative is expert estimation. The expert is currently at least as good as the models, and our idea is that there is much greater potential to improve the expert than to improve the models.

"But when I looked at the way that experts were working, I saw that we, the researchers and the organizations, looked at expert estimation as a black box. I have an expert, I get an estimate. Yet it is obvious that you can support experts with better training and better input data. It was also obvious that there were a lot of biases in expert estimation. We also discovered that having some irrelevant or misleading information strongly impacted their estimates."

"Can you give me an example of that?"

"Suppose that a software developer knows what the budget of the client is. Even if it is unrealistically low, and not based on any knowledge of what it really costs, but is a budget based only on how much money the client has, knowing this budget makes it nearly impossible to be realistic about what the project will really cost. The strange thing is that you think that you're being realistic. You manipulate your thinking to believe that it's possible to work within the client's budget.

"There have been many studies on the human inability to separate what should be and what is realistic. Our idea is to remove those elements that make the expert unrealistic. There is still so little awareness of the impact of these things in the software industry."

"What makes software engineering such a difficult area for effort estimation?"

"It's very useful to compare it with the construction of buildings. Building contractors have very good estimation models, using very few variables. In some construction contexts they have an estimation error of only 3 to 4 per cent. In software engineering, we have struggled for years and the average error is still 30 to 40 per cent. I think that there are three main differences. First, we do new things all the time, which means that the relevance of our old data is not obvious. If you change what you are building, the old data is not as useful. Second, in software engineering the core relationships are not stable. In construction, the number of blocks has a stable relationship to the cost. The same is not the case in software engineering. The relationship of size to cost is not stable, and if you don't have a stable relationship then it's very difficult to develop models.

"Third, we have what you may call highly specific information. A project manager may have one engineer named Ole who is highly skilled. If he is doing the work it will take one month, while if you have a normal-speed person with no experience, it will take one year, for example. This information about Ole is very hard to put into an estimation model, because a model has to be general. There is no room for highly specific information in an ordinary model, unless you have an extremely high number of historical data points."

"What sort of information typically goes into a model?"

"The size in some sense is the main input: what you believe will be the number of lines of code, or how many function points the software has. Then you might also have a measure of complexity. There are many different attempts to measure complexity and also size. If you have many measures of a quantity, this is an indication that you don't have a *good* measure. If you had a good measure, everybody would use that measure! In construction, square meters are a good measure of size. In software you have no measure of size. You don't know the number of lines of code when you start. You have to estimate the lines of code, and there are studies showing you're even worse at estimating the number of lines of code than the number of work hours."

"When did you start working on judgement-based estimation?"

"It's hard to say when I actually started, but in 1999 there was an important change when I shifted my main focus to judgement-based estimation. In the beginning I combined my focus on the models and expert judgement, but over time I worked more and more on judgement and less and less on models. However, I still do some work on estimation models."

"Did other experts in the field think you were crazy when you started doing this work on expert judgement?"

"No, I think that nearly all of them said that this was a valid and important topic. It's very easy to accept in a way, because industry mainly uses this type of estimation process. The problem is that if you are very good at building models and using statistics, but you have never taken a single course on psychology or organizational theory, then the threshold to starting research on expert judgment is very high. I had to invest many years to gain competence."

"How did you do that, without a degree in psychology?"

"It started with a researcher at the University of Oslo, Professor Geir Kirkebøen. He combined a background in informatics with a strong focus on psychology. I started discussing this with him, and he gave me a lot of articles I could read, and I started writing a paper with him. He was sort of my mentor in the beginning. Gradually I started reading more and making more contact with people in that environment. At the University of Oslo there is a very strong research group on human judgement, led by Professor Karl Halvor Teigen. I still have a strong link with them. We have common seminars together. I was lucky that there was such a strong group in that topic, so close to Simula."

"How did Simula impact your ability to do this research?"

"Simula has made this possible because they have a long-term perspective on things. They have supported and made my large-scale experiments possible. And there's a great degree of freedom here. I've been allowed to focus more on research, with less administration and less lecturing. Also, my studies have been accepted more readily because they have been realistic studies. If I only had psychology students as experimental subjects, it would have been much harder to get them accepted in both industry and academia."

"Let's talk about the experiments you are conducting. You say that these are the first randomized controlled experiments in software engineering. How many have you done so far?"

"If you only count those with 100 per cent realism, where the participants did not feel that they were participating in an experiment in any way, and they thought they were doing 100 per cent ordinary work, we have done three randomized controlled large-scale experiments. Of course, there are many other large-scale industry experiments with a randomized treatment, but in those, the participants knew that they were part of an experiment. So it depends on how you define it."

"Does it make a significant difference when people know they are part of an experiment?"

"The problem is that we don't know. I have the perception that people are working very close to normal even when they know they are part of an experiment, but we don't know that for sure. That's a topic in itself, really! You always have this threat when the participant knows it is not real work. Even though they get their ordinary payment, even though it's an ordinary context, they may work slightly differently."

"What kind of tasks did you ask people to do in these three experiments?"

"In the first experiment they had a specification, a description of the functionality of the system. All of them were Norwegian companies in the first experiment, and they got the same specifications. The task was to estimate and give us a bid for this work. We had a treatment, where they got the specifications in a different sequence. One group received a full specification at the outset and the other received a reduced specification with less functionality, which we then added onto. This actually had a large effect. People starting with the smaller version who then got a new version with added functionality ended up much higher. We used this result to recommend changes to common bidding practices, and also as an input to our understanding of the mental steps of expert estimation."

"In cost, or effort, or both?"

"It was in cost, because we did not ask for the effort estimates. It is to be expected that the effort estimate would be higher, too.

"Also, we noticed that the ones with the least relevant experience had the lowest estimates. You would expect those with the highest competence to have the lowest estimates, if it was a rational world out there. But this means that if you select a company based on a low estimate, you will normally get the least competent one. When the company has more experience, and knows more about the problem, this will most likely lead to higher estimates and higher bids, which is sort of the opposite of what you would hope for.

"The second study was a follow-up on that first one, where we also had a difference in the sequence. That one was with outsourcing companies in India and Eastern Europe."

"That sounds very interesting, with the whole idea of the cross-cultural differences. What differences did you find?"

"We're still in the phase of summarizing the cultural findings. The reason is not that we don't have the results, but that things with culture are very complex. We have data on a few companies in India, but this is a country with one billion inhabitants. It's hard to say what is typical for Indian companies, when there are so many companies and we are not really highly competent in cultural studies.

"Nevertheless, there are differences, and they are very strong, in our experience. The over-optimism seems to be much stronger in Indian and Pakistani companies than in Eastern European countries. That's our experience based on 70 to 80 companies."

"So in theory that should give them an advantage in winning projects but a disadvantage in terms of complying with them?"

"Yes."

"What was your third study?"

"That's the most recent one, where we also had outsourcing companies. We put in a lot of misleading information, like the example I gave you earlier with the budget information. We tell them that the budget is so and so, but that this budget will be extended if necessary and they should not let their effort estimates be impacted by that. Still, they are strongly impacted.

"That study is the first one studying the effects of irrelevant and misleading information in a real-world context. Even in psychology, where they study so-called 'anchoring' effects, like the budget value, they haven't done studies like this in a professional context. Either they have asked students—perhaps something like 95 per cent of studies in psychology are done with students—or they have asked professionals in a very artificial context. This is the first one that is both asking professionals and in a realistic context."

"What kind of results have you gotten? Is it still too early to say?"

"No, that one has been summarized. The main finding is that some effects, like the length of a specification, are much stronger in a lab context than in a real-world context. Some effects are almost removed when you go to a professional context.

"For example, we looked at whether if you give the subject exactly the same content, but through wider margins and larger fonts you make it much longer in the number of pages, does this give you higher estimates? In a laboratory context, an artificial increase in the length does lead to higher estimates. But it's a very small effect in real life.

"Also, we studied effects of budget information, and that is maybe the most amazing. We said that the relevant work should be completed in a three-week period. This should rationally lead to more use of effort, because you have to put more people on the project and coordinate more people, in order to complete the project in a squeezed time frame. But what happened, both in the lab and just as much in the real-world context, was that the effort estimates got much lower. The reason is easy to explain. It was over-optimism. You hope that you will be able to complete this in three weeks with the same number of people. You manipulate yourself into believing this will not cost more. This is again most likely explained by our inability to separate wishful thinking from what is realistic. It would be very convenient if it was possible to do this work in three weeks with only one or two people, and after a while you start to believe that it is realistic."

"Can you tell us about some of the experiments where you would actually observe people?"

"Those are lab experiments in more or less realistic settings. In one study we studied estimation teams, because estimation work is often done with teams. We video-recorded a lot of estimation teams and transcribed the discussions. I have a PhD student who is analyzing these transcriptions.

"How do people go from an understanding of the problem into a number? How does this step happen? Maybe the most striking observation was that magic step. The groups would discuss the specifications of the project in detail, this and this and that, and then there's a moment of silence, and then out comes a number, say 18. Normally it comes out without any explanation of why 18 and not 17 or 11. Then another guy will say, I think it's 24, and then they will start negotiating. Not explaining, but negotiating, and they end up in the middle. Or if the senior person is the one with the highest or lowest estimate, maybe they will go with that number.

"This has given us insight into how it's actually done, but it also shows us what we don't know. Some steps are very explicit, but the most important step, the number generation, is very implicit and very intuition-based. They are not able to describe how they end up with the number 18. They may rationalize and say something about it, but if you push them a little bit you notice that this step is unconscious. They don't know why they think 18 is right. It's sort of a feeling."

"It's striking to me that you describe it as a process of negotiation. Can you exploit that fact? Maybe you want to acknowledge that it's an economic rather than a model-based process?"

"I think what I would say is that combining different opinions works, whether it's through an average value or a negotiation. Perhaps the strongest result in forecasting, which is also supported by our research, is that a combination of independent sources gives you a more realistic view of the outcome. Negotiation is one way of combining opinions. It seems to be not very important how you combine them, as long as it's a meaningful combination. All of these methods are good, and they are better than having only one opinion."

"How does it compare if you put people together on a team versus doing it separately?"

"We have done a number of studies on team-based effort estimation, and also on combining estimates without teams. It seems that, as you indicated, making them start independently and then meet and discuss their estimates is better than having a mechanical combination of individual estimates, or having them discuss the estimate without first writing an independent document. Otherwise it can easily happen that the most dominant person says a number, and this number is a reference for all other estimates, so it's really one estimate even though you have four people.

"Also we have noticed that if people meet, one of them may remember a task that two of the others have forgotten. Then this task will become part of the common estimate. Sometimes people may end up with the same effort estimates for the wrong reasons. If you meet, you will notice that each of you missed one activity, and so the team estimate may end up higher."

"Do you have any other recommendations for avoiding over-optimism?"

"I think the most important, the one I like the most at least, is the one about assessing the uncertainty of an effort estimate. In project management and software development effort estimation, you typically ask for the minimum, the maximum, and the most likely. We have shown that the minimum and maximum are much too close to the most likely.

"One of the methods we have developed makes people more realistic about the minimum and maximum. It's simple. Instead of asking for the minimum and maximum, you ask: What has been the distribution of estimation error previously? You remind people about the history. When you ask about the minimum and maximum, people look forward. Looking forward makes people more optimistic. Looking backward on the history makes them more realistic. We ask them to make a distribution of estimation error, looking back, say, over 50 projects. They are actually able to do this. It is easy to use this as a platform for estimating the uncertainty of this project, assuming you accept that the history is relevant. It is not always very accurate, but it is much better than not using the history."

"Have your clients ever done this sort of exercise before?"

"No. And if you look at textbooks on project management, for example, they assume that people are able to give meaningful minimum and maximum estimates without any support. That's a huge shortcoming of these textbooks. They assume that people are realistic about the uncertainty of their efforts, while estimates show again and again and again that they aren't. That makes me really believe that these authors are not aware of the literature in psychology on overconfidence, because they never mention it.

"This is not new knowledge; it has been around for at least 20 years. This is a very good illustration that professional management, even though it's about people, is not very people-oriented, or at least not based on an updated understanding of human biases, for example."

"Do Simula's industry partners use some of these ideas?"

"I would really like to have a way of documenting what they actually do. I have a strong feeling that they use the results, and some of them have said they do, but I have no objective way of measuring whether they are applying it meaningfully and not misusing it. I get emails frequently from people who say they are using our results. But that is only emails and personal communications. It would be much nicer if I had some hard facts.

"The fact that we get invited to have seminars at Norwegian and European companies shows that they at least think our work is relevant. I present my results at perhaps 10 or 15 industry seminars a year, and I have many more invitations than I can accept."

"Do you see any possibility for a spin-off at some point?"

"We have already had one spinoff company, led by one of my PhD students, called Project Economics. And there is a new one recently started called Wasteless, which is also based on some of the ideas here. The potential is great, but it's more about my time, really. If the results were neatly packaged and we could sell them in a box that would be easier, but this area is very expertise oriented."

"So you yourself are not interested in starting a spinoff?"

"Not now, at least. It's big fun to work with the software industry. It's not that I don't like it, but I like doing research even more."

"In the workshops that you lead, have you started to see changes in how prepared people are?"

"Yes. I hear industry people at conferences talking about anchoring effects and such things. That's amazing because five years ago, I heard nobody talking about these psychological effects. It's hard to say with strong confidence that our work is the reason, but that's my personal opinion! There has been a change in the awareness of human biases and effects like anchoring, that's for sure. It's no wonder, because there must be several thousand software engineers that have listened to our presentations, and I have written in magazines and other places[1]."

"Is there anything else that you would like to add?"

"The most important point is the obvious fact that software is developed and estimated by people, but our software engineering education is strongly oriented toward technical things, for historical reasons. Software engineering started as this subdiscipline of mathematics, where there was no room for psychology. It's gradually changed, but there are some obstacles that stop us from being rational. We accept that programming is done by people, but our research is ignoring it, even when the models have failed, as in estimation research. I noticed this and had to make a choice. Should I focus on something others and myself had worked on for years without success, or should I invest three or four years in building up competence in psychology so that I could combine it with research on effort estimation? My hope is that more people will not only accept that psychology and human biases are important, but will invest time in building up competence and alliances with psychology research. So that's my main message here."

[1] Among other things, Jørgensen writes a monthly column in the Norwegian Computerworld magazine.

26

SOFTWARE DEVELOPMENT EFFORT ESTIMATION — DEMYSTIFYING AND IMPROVING EXPERT ESTIMATION

Magne Jørgensen and Stein Grimstad

Magne Jørgensen · Stein Grimstad
Simula Research Laboratory

Magne Jørgensen
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_26,

PROJECT OVERVIEW

Better Estimation of Software Tasks

Inaccurate estimation of software development effort is one of the most important reasons of IT project failures. While too low effort estimates may lead to project management problems, delayed deliveries, budget overruns and low software quality, too high effort estimates may lead to lost business opportunities and inefficient use of resources. These IT project problems motivated the BEST (Better Estimation of Software Tasks) project at Simula Research Laboratory to conduct research with the goal of improving effort estimation methods.

Scientific challenges

The project's main focus is to improve judgment-based effort estimation (expert estimation), which is the estimation approach most frequently used by the software industry. We argue that a better understanding of the mental steps involved in expert estimation is necessary to achieve robust improvement of software development estimation processes. A great challenge when studying expert judgment is to understand the unconscious steps involved. To study these steps, we make use of multidisciplinary competencies, especially psychology and software engineering, and financial resources enabling studies in realistic software development effort estimation contexts.

Obtained and expected results

In this chapter, we present the main results of the project from its beginnings in 2001 until today and describe how the project has benefited from being an integrated part of Simula Research Laboratory, especially through the opportunity to conduct large-scale, controlled experiments in field settings. The chapter includes practical results on how to improve estimation methods, scientific results leading to a better understanding of the mental processes involved in judgment-based effort estimation, and our innovative research methods in the field of empirical software engineering. Results of the BEST project are currently in use by several software companies, as well as by researchers in forecasting and psychology, and they are also included in project management and software engineering textbooks. The BEST project plans to continue its multidisciplinary effort with the goal of constructing and evaluating better models of the mental steps involved in judgment-based effort estimation. An improved model of these steps will enhance our ability to accurately estimate software development effort and to predict when different types of estimation methods can be expected to deliver accurate effort estimates. This will in turn lead to a reduction of the number of IT project failures and make better use of scarce financial and human resources.

SOFTWARE DEVELOPMENT EFFORT ESTIMATION — DEMYSTIFYING AND IMPROVING EXPERT ESTIMATION

26.1 Motivation

Industrial Relevance

The main determinant of many types of software-related investments is the amount of development effort required. The ability of software clients to make investment decisions based on cost estimates is consequently strongly tied to the software providers' ability to estimate the effort accurately. Similarly, the ability of project managers to plan a project and ensure efficient development frequently depends on accurate effort estimates. The importance of accurate effort estimates is illustrated by the findings of a 2007 survey of more than 1,000 IT professionals[1]. The survey reports that two out of the three-most-important causes of IT project failure were related to poor resource estimation, that is, inaccurate effort estimates.

The survey response is understandable. A review of estimation accuracy studies [64] reports that software projects expend, on average, 30 to 40 per cent more effort than is estimated. Software projects experience severe delivery and management problems when plans are based on overoptimistic effort estimates. The negative effects of overoptimism are accentuated by software bidding rounds in which those companies that provide the overoptimistic effort estimates are more likely to be selected (the "winner's curse" effect, see [42]), and strong overconfidence in the accuracy of the estimates–for example, 90 per cent confidence intervals (minimum-maximum intervals) of effort only include the actual effort 60 to 70 per cent of the time [58]. Not only can effort underestimates lead to problems with software projects (both for the provider and the client), but effort overestimates are also problematic. Overestimates may, for example, lead a provider to lose bidding rounds and clients not to start a project that would have been a good investment. Solving the problems related to inaccurate effort estimates would clearly improve the use of scarce financial and human software development resources.

While we cannot expect zero estimation error, due to the high complexity and inherent uncertainty of many software projects, there are reasons to believe that

[1] See: http://certification.comptia.org/project

the estimation accuracy improvement potential is high. Reasons for believing this include the current strong bias towards overoptimism (cost overruns are far more typical than cost underruns), the high degree of inconsistency in the estimates (effort estimates for the same work can vary greatly, even when repeatedly estimated by the same developers), the limited collection and use of historical data (poor track record of learning from experience), and the lack of evidence-based selection of estimation methods. We discuss our results related to these reasons in more detail in the results section of this chapter.

Previous Research

Software engineering researchers have been addressing the problems of inaccurate effort estimation in software development projects since at least the 1960s; for example, see Farr [11]. Nearly all of that research has focused on the construction of formal software effort estimation models. Model types that have been developed and evaluated include those founded on regression analysis, case-based reasoning, classification and regression trees, simulation, neural networks, Bayesian statistics, lexical analyses of requirement specifications, genetic programming, linear programming, economic production models, soft computing, fuzzy logic modelling, statistical bootstrapping, and combinations of one or more of these models. In a review, we identified 184 journal papers that introduced new variants of formal models for software development effort estimation [53]. This corresponds to a per centage as high as 60 per cent of the total number of journal papers on effort estimation. Several formal estimation models have been included in commercially promoted tools. A survey by Moores and Edwards [69] found, for example, that 61 per cent of the IT managers in the UK had heard about at least one of these tools for software development effort estimation. The use of formal estimation models is also typically promoted by software process improvement frameworks and in educational readings on software engineering.

Yet in spite of the extensive research into estimation models, the high degree of availability of commercial estimation tools that implement the models, the awareness of these estimation tools, and the promotion of model-based estimation, software engineers typically use judgment-based methods[2] ("expert estimation") to estimate effort [21, 22]. At Simula Research Laboratory, a recent survey by Nils Christian Haugen (work in progress) suggests that the use of formal estimation models has declined, rather than increased, over the last five years.

[2] We find it meaningful to categorize judgment- and model-based effort estimates on the basis of the type of mental process applied in the "quantification steps" of the estimation work, that is, the steps in which an understanding of the software development estimation problem is translated into quantitative measures of the required effort, such as the number of work-hours most likely needed. If the quantification steps are based on tacit, intuition-based processes, the process is categorized as *judgment-based.* If the quantification steps are mechanical, the process is categorized as *model-based.* There will be a range of different estimation processes belonging to each of the categories, that is, neither expert judgment- nor model-based effort estimation should be considered simply as "one method". Software development estimation models typically take expert judgments as input, that is, both estimation approaches depend, to some extent, on subjective input.

Why Improve Judgment-based Effort Estimation?

Our research is based on the assumption that research on judgment-based effort estimation processes (sometimes in combination with formal estimation models) would be an efficient use of research resources when used to improve effort estimation accuracy. Reasons in support of this assumption include:

1. Even the use of unstructured and unsupported judgment-based effort estimation seems, on average, to lead to more accurate effort estimates than the use of sophisticated formal models, as reported in our systematic review of empirical studies comparing expert and model estimation accuracy [27]. The introduction of more structure, such as the use of structured group processes and an experience-based estimation checklist, consequently has the potential to substantially improve judgment-based effort estimates (see, for example, [2], for general forecasting evidence on the benefits of several types of expert judgment process structures and support). In [27], we report that the following two characteristics of the software development context may point to inherent, essential advantages of judgment-based effort estimation compared with model-based effort estimation:

   a. Software development domain experts typically possess highly context-specific information of importance that is not part of the model. This situation has been found to favour judgment-based estimation over model-based estimation [13, 76].
   b. Essential software development relationships, such as between effort and size, seem to be unstable [7]. As pointed out in [72] judgment-based forecasts worked better in unstable situations, while the models performed better during periods of stability. Software development is characterized by unstable, dynamic conditions with frequent changes of problems to be solved, new teams, new clients, and new tools.

2. In comparison with that conducted on formal effort estimation models, there has so far been very little research on improving judgmental effort estimation processes. As an illustration, we found that only 15 per cent of the journal papers on software development effort estimation analyzed judgment-based effort estimation [53]. Among these papers, we found none that aimed at an *improvement* of judgment-based effort estimation.
3. Overall, the software industry seems to be more willing to accept and use judgment-based estimation methods. We have, for example, observed repeatedly that software projects that officially apply a formal estimation model, in reality use the model as a disguise for judgment-based estimation [44]. This low acceptance of model output in some contexts is observed in several disciplines and should not be surprising. In [19], for example, analysis (corresponds to model-based estimates) is described to have the characteristics of “low confidence in outcome, high confidence in method”, while intuition (corresponds to judgment-based estimates) is described to have the characteristics of high confidence in outcome, low confidence in method (see also [10] for similar findings). An important reason for the rejection of output from formal software development effort estimation

models, in addition to their low accuracy, may be that the experts' highly specific knowledge, for example, specific knowledge about the particular developers who are supposed to do the work, frequently cannot be included properly as model input. Understandably, it is not easy to trust an estimation method that is unable to make use of clearly relevant information and that does not "feel right". However, it is hardly possible to unite highly specific knowledge with the general relationships that are required by formal models.

4. It is difficult to avoid relying on software professionals' judgment when estimating effort, even when using estimation models. Improvement of the judgmental processes is consequently of importance in, perhaps, all real-life estimation situations.

A discussion between the first author of this paper and Barry Boehm (perhaps the most well-known estimation model builder) on the advantages and disadvantages of model- and judgment-based effort estimation can be found in [36].

The Objectives of the BEST Project

The Better Estimation of Software Tasks (BEST) project, launched at Simula Research Laboratory in 2001, built on a 2000 project started at the University of Oslo. The BEST project had and still has the following three main, interconnected objectives related to judgment-based effort estimation:

O1: Better Understanding (Theory-building). The main goal here has been to develop an evidence-based, "constructive" model (theory) of judgment-based effort estimation. A constructive model in this context should be interpreted as one that is useful for the proposal (construction) of improved judgment-based effort estimation processes and principles. Such a constructive model should, for example, be able to support the design and selection of estimation methods and predict when an estimation situation is likely to lead to inaccurate effort estimates. Moreover, our model of judgment-based effort estimation should also lead to an improved understanding of quantitative judgment in general and make a positive impact on research in other domains with similar types of judgments.

O2: Better Practices (Improved estimation guidelines and processes). The main goal here has been to develop methods and principles that have the potential to generate substantially more accurate effort estimates than currently occurs for software projects. This may include the construction of new methods for effort estimation, methods that combine the advantages of models and judgments, and methods for the improved selection of estimation methods. The methods and principles should, preferably, be based on robust theory on human judgment.

O3: Substantial Industry Impact (Transfer of results). The main goal here has been to disseminate improved, validated in-the-field, estimation guidelines and processes to the software industry. The methods and principles should be in a format that makes them adaptable to a variety of industrial contexts.

26.2 Research Results

This section contains brief descriptions of what we consider our main research contributions on judgment-based effort estimation. The results presented in the following sections on improved knowledge about state-of-practice and improvement of research methods provide an important basis for achieving the three objectives (O1, O2, and O3) of the BEST project. The section on improved understanding of judgment-based effort estimation presents results with a focus on O1 while results with a focus on O2 is presented in the improved judgment-based effort estimation processes section on page 392. The results related to O3 are presented in section 26.3.

The actual studies and additional details about the results and proposed estimation methods can be found in the referenced studies.

We have not included our results on formal estimation models in this paper. Several of these results, however, particularly the uncertainty assessment model proposed in [55] and the regression-towards-mean adjustment of analogy-based models proposed in [46], have been subject to subsequent study by other researchers and/or inclusion in commercially available estimation tools, that is, they have had an impact on both research and practice. Neither have we included our contributions on Evidence-Based Software Engineering [59, 8, 47].

Improved Knowledge about State-of-Practice

A deep understanding about state-of-practice is necessary to focus the research on essential topics and to comprehend the potential for improvement. Our main contributions documenting the state-of-practice are related to:

1. Surveys of estimation accuracy, estimation processes, and reasons of estimation error.
2. Disclosure of the low level of trustworthiness and the survey-method problems of the most frequently cited report documenting the state-of-practice in estimation work, that is, the Standish Group's CHAOS report.
3. The first documentation of the high level of inconsistency in software professionals' effort estimates.
4. The first documentation of the high level of overconfidence in the accuracy of software professionals' effort estimates.
5. The first in-depth study comparing top-down and bottom-up estimation processes.
6. Studies on group-based and mechanical combination of effort estimation.
7. The first review comparing the estimation accuracy of models and expert judgment.
8. Developing the currently most comprehensive database and review on estimation research.
9. The first field experiment on the (lack of) learning from estimation 'lessons learned' sessions.

Through our own surveys on estimation practices [66] and reviews of other surveys [64], we have documented a 30 to 40 per cent estimation error, on average, with no indication of improvement over the years. We also have documented the infrequent use of formal estimation models and the typical reliance on bottom-up, judgment-based effort estimation [31]. We report reasons for estimation error and indicators of high/low estimation error in [32, 49]. Two main contributions from our studies on the reasons for estimation errors are: Current estimation error analysis practices are hampered by too strong a focus on direct reasons, inclusion of limited perspectives, and the application of simplistic cause-effect models; and the best indicator of overoptimistic effort estimates is the previous level of estimator's overoptimism, but even that indicator is not optimal, that is, it seems to be difficult to predict when an effort estimate is overoptimistic. We survey the practice of estimation error analysis in [16] and find that the current practice presents severe shortcomings, for example, there is a lack of precise estimation terminology.

By far, the most frequently cited survey concerning effort estimation accuracy is one referred to in governmental reports as well as research papers. It was conducted by the Standish Group in 1994 and is known as the CHAOS report[3]. It reported an average cost overrun of 189 per cent. This cost overrun number has been used for several purposes, including excusing poor estimation work. In [48], however, we show that the results reported by the Standish Group are not trustworthy. We argue that the Standish Group's results, which deviate from all other studies on estimation error, are likely to have been caused by a strongly biased sample of projects. More recent reports from the Standish Group, on various topics, seem to exhibit many of the same survey method problems. This illustrates the need to increase awareness of proper survey methodology among researchers and software practitioners.

It is reasonable to expect that the effort estimates of the same project given to the same software professional, with but a few weeks in between, would not be identical; nevertheless, the size of the difference in the estimates (the level of inconsistency) we found turned out to be amazingly large [15]. The median difference between the estimates of the same tasks by the same software professional was as high as 50 per cent. An implication of our study is that low estimation accuracy is, to a substantial extent, caused by a lack of consistency in the effort estimates. These results demonstrate the low robustness of some judgment-based estimation processes, and that increased consistency should be considered as an important method of improving estimation accuracy. The high degree of inconsistency also explains, to some extent, why it is difficult to predict overoptimism, that is, the results reported earlier herein.

Our studies on overconfidence in the accuracy of software development effort estimates are the first in field settings [58]. They document not only the high level of overconfidence–for example, that the figure "90 per cent confident" to include the actual effort in a minimum-maximum interval corresponds to "hit rates" of 60 to 70 per cent–but also demonstrates serious problems with the current guidelines for uncertainty assessment, as prescribed in common textbooks on project management. As an illustration of the problem with current approaches based on effort prediction intervals, we found only minor differences in effort prediction intervals when requesting 99 per cent, 90 per cent, 75 per cent, or 50 per cent confidence to include the actual effort, that is, that the participants provided almost the same minimum and

[3] http://www.standishgroup.com/

maximum effort values when they were asked to be 50 per cent certain, compared with 99 per cent certain to include the actual effort. Results from our studies on overconfidence, particularly our results providing new insight into the reasons for the overconfidence, have been included in the paper "Judgmental forecasting: A review of progress over the last 25 years", *International Journal of Forecasting* [61].

Expert estimation processes are frequently described as either top-down or bottom-up, that is, a process in which either the effort estimate is derived from properties of the project as a whole (top-down) or derived from a decomposition of the project into activities (bottom-up). Our video-recording of estimation teams' discussions when applying these processes resulted in a documentation of the estimation processes that they used along with insight as to when the processes led to the most accurate estimate [33]. The main observation was that the top-down strategy offered the best choice when the estimators brought to bear experience gleaned from a similar project. This new project, however, had to be quite similar for the analogy to be useful. If the software developers could identify projects that were only somewhat similar, then the bottom-up process produced more accurate results. The video-recordings and our results are now subject to further analysis of the interaction processes in estimation teams, as part of a PhD thesis in education and learning.

We have completed several studies on combination-based effort estimation [63, 67, 65, 52]. These studies document the benefits of common as well as more innovative combination strategies. Of particular interest are the results on the use of the estimation method commonly applied in agile projects, such as "planning poker". While many studies in other domains observe an increase in risk-willingness in teams (for example, see [4], we did not find this effect in group-based effort estimation. Instead, we found indications of the opposite, that is, that group dynamic effects led to increasingly higher effort estimates Our study on different strategies of combination of effort prediction intervals [52] is the first in a software engineering context. The results suggest that a discussion-based combination of prediction intervals should be used instead of mechanical combinations of individual prediction intervals. The overall result of our studies on combination of estimates from different sources, either mechanically or as structured group-discussions, finds that combination of estimates typically improves the average estimation accuracy compared with effort estimation based on one source only. This corresponds to results from other domains.

Our observation (see the review in [27] that formal estimation models did not produce more accurate effort estimates than those based on expert judgment is surprising in light of related research in other domains. A meta-analysis of comparisons of models and judgments is provided by Grove et al. [17] who found that mechanical predictions of human behaviours are equal or superior to clinical prediction methods for a wide range of circumstances. Dawes, Faust, and Meehl [5] emphasize the following two factors that underlie the observed superiority of statistical models: Models are consistent (the same input always leads to the same conclusion), while experts are inconsistent; and models ensure the contribution that variables make to a conclusion is based on the variables' actual predictive power and relationship to the phenomenon of interest. There may, however, be essential differences between most of the previously studied domains and software development that can explain the lack of advantages to using effort estimation models in software development contexts. As described earlier, there are at least two characteristics of the software development context that may point to inherent, essential advantages of judgment-based effort estimation, that is, that software professionals typically possess highly context-specific information that is not part of the model, and that essential software development relations seem to be unstable. These two advantages may compensate for the possible disadvantages of judgment-based estimates, for example, a higher degree of inconsistency and improper weighting of variables.

As part of building a publicly available database of all software development effort estimation research[4], we categorized the properties of previous estimation research [53]. The main contributions of that work are to support researchers with topics for future research and show where relevant research can be found. To date, more than 100 PhD students and researchers have used this database. The feedback we have received indicates that it has saved many of them considerable work, and that the quality of their research has been enhanced through the information provided by our database.

Learning from experience through lessons-learned sessions is a frequently proposed strategy to improve estimation accuracy. We evaluated how effective such lessons learned were and discovered a surprising result: They had no effect on estimation accuracy and effort uncertainty assessment realism [43]. Based on qualitative analysis of the data, we propose that the problem is rooted in the difficulty in assessing how much one has learned from experience. If, for example, a software developer learns from experience, but is overoptimistic about how much he or she has learned, the effort estimates will continue to be overoptimistic. We also found evidence suggesting that estimation learning on behalf of other software developers is easier than learning from one's own estimation experience. We plan to work on better estimation learning processes as a follow-up to these, somewhat surprising, results.

Improved Understanding of Judgment-Based Effort Estimation

Robust improvement of effort estimation practices is facilitated by an improved understanding of the mental processes involved in judgment-based effort estimation. Without a comprehensive understanding of why and when different types of judgment-based effort estimation methods can be expected to be accurate, the design and selection of estimation method will prove problematic. Our contributions towards a constructive model of expert judgment include:

1. The first studies documenting the importance of unconscious processes in judgment-based software development effort estimation.
2. The first studies on how much, why, and when irrelevant and misleading information impacts the unconscious processes involved in judgment-based software development effort estimates.
3. The first studies on how the format of requesting effort estimates impacts the level of optimism.

[4] www.simula.no\BESTweb

4. The first studies on how software professionals select and change estimation strategy.
5. The first attempt to synthesize existing results from various disciplines into a constructive model of judgment-based effort estimation.

Through video-recording of an estimation team's discussions, use of think-aloud protocols, interviews with software professionals, and studies in which we demonstrate that software professionals do not notice when and how much they are impacted by irrelevant information, we document that essential parts of the estimation processes are based on unconscious processes [33, 29]. An important implication of this finding is that we cannot simply ask software professionals or use think-aloud protocols to examine the real mental processes, but instead will need to rely on controlled experiments.

Another consequence of the unconscious nature of essential steps in judgment-based effort estimation is that the estimates are easily impacted by irrelevant and misleading information. We have conducted more than 30 laboratory and in-the-field experiments documenting this, such as studies on how the estimates and bids are affected by early and underestimates based on limited information [56, 37], clients' unrealistic price expectations or insufficient budgets [54, 39], future, promising opportunities [39], the use of loaded words, such as describing the same task as a major or minor extension [39], and inclusion of text not relevant for the use of effort [39]. Since the software professionals are unaware to what extent they are impacted by misleading or irrelevant information, it is very difficult, if not impossible, to adjust for the impact even when one knows that the information is misleading or irrelevant (for example, see [40]. Our most recent study on this topic [41] is, as far as we know, the first that compares the effects of misleading and irrelevant information in field settings with corresponding effects in laboratory studies, regardless of domain. This type of field study is, in our opinion, essential to examine the importance–as opposed to merely the existence–of a phenomenon. We find in our field studies, for example, that the impact from some types of irrelevant information, such as the impact from the length of the specification, is present mainly in laboratory settings, while the impact from other types of irrelevant effort information, such as the clients' unrealistic cost expectations, is lower but still important in field settings.

In a recent study, we examined the importance of the format of the estimation request [45]. In it, we report four experiments suggesting that the request format: "How much of Y can you complete spending X work-hours?", which is a format sometimes used in agile projects and by clients, under certain circumstances leads to more optimistic effort estimates compared with the traditional request: "How much effort will it take to complete Y?" An explanation of this finding, may prove relevant when trying to understand the mental processes involved in judgement-based effort estimates.

It is unlikely that software professionals use only one strategy when estimating effort. Instead, they may rely on a toolbox of strategies and an assessment of the fit of individual strategies to the current situation. Recently, we reported that there is a large individual variation in selection in situations for which there is no information clearly in favour of one particular strategy [26]. We also found that the following three strategy selection factors play important roles in situations for which relevant, historical data are available: individual strategy preference (prior beliefs), estimation surprise (large estimation errors), and aggregated accuracy information of a strategy in the current context. In the study, we found that estimation surprise was frequently required to make the developers change their estimation strategy, even when one strategy was clearly better than another. We also found that the strategy applied toward unrelated tasks immediately before the estimation work occurred did impact the choice of estimation strategy (a so-called priming effect). For example, the completion of unrelated closest-analogy tasks immediately before led to more use of the closest-analogy strategy in the estimation work.

An important vision of the BEST project has been to synthesize existing results from various disciplines into a constructive model of judgment-based effort estimation and add our own results wherever we find research gaps or a need for more evidence. A constructive model in this case means a model that will be useful for predicting and improving expert judgment-based effort estimates. While we are still some steps away from reaching this goal, we have made a couple of preliminary models [29, 24] and identified research results from other domains, particularly the "selective accessibility" theory presented by [70] and the "construal level" theory presented by [75].

Improved Judgment-Based Effort Estimation Processes

Our research on judgment-based effort estimation has led to several documented improvements of the effort estimation (including bidding and uncertainty assessment) processes. The main process improvements are, in our opinion:

1. Improved estimation terminology.
2. An improved process for analyzing the estimation error.
3. A simple, yet realism-increasing, evidence-based method for the assessment of effort estimation uncertainty.
4. Evidence-based guidelines and principles of judgment-based effort estimation on preparation of estimation information, when to use models, how to combine, how to assess the uncertainty of an effort estimate, how to avoid common estimation biases, and so forth.
5. Guidelines on how to avoid bidding based on overoptimistic estimates.

Starting with our work [35] and further elaborated in [16], we show that the current use of estimation terminology in research papers, software engineering textbooks, and industry practice is confusing and does not enable the types of estimation error analyses we would like to complete. As an illustration, the term "effort estimate" is used, variously, to denote the most likely use of effort, the median effort, the planned effort, or the effort used to price the project. To add to the confusion, this inconsistent use of the term typically appears with no definition of the intended meaning [16, 56]. This lack of precision in the terminology is unfortunate both because a measure of estimation error will be difficult to interpret and there easily could be an unfortunate mix of different concerns in the estimation process. To reduce these estimation terminology problems, the concern when estimating effort should be an assessment of realistic use of *work effort only* (not for example planning or pricing) and the meaning of an estimate should be clearly defined. The estimated effort should then be used as *input* to planned effort (whereby the main concern should be project control and efficiency) and price (whereby the main concern should be profit on short or long term) (see [51]. Not separating realism, planning, and pricing concerns implies that 'wishful thinking' may dominate on the cost of realism [9, 12]. Related to the clarification of what is meant by an effort estimate, we propose the use of pX-estimates, where X is the likelihood not to exceed the pX value. For example, the intended meaning of a p50-estimate is that this type of estimate will be exceeded about 50 per cent of the time. This type of probability-based estimation terminology has been proposed earlier, see, for example, [6]. One problem arising with previous proposals, however, has been that they provided no practical way of producing the probability-based estimates. We propose a simple, yet accurate process for this based on a combination of estimation of most likely effort and use of the distribution of estimation error of similar projects. The process has been evaluated with success in several field settings [30, 34]. An important property of the pX-based terminology is that it enables a practical separation of concerns, for example, that the p70-estimate could be the planned effort and the p80-estimate the basis of the budget or price to client.

In several papers, for example, [56, 23, 14], we discuss problems with current processes and measures of estimation error evaluation. We show, among others, how effort estimates impact the actual use of effort and, for this reason, make it difficult to evaluate the estimates' accuracy. There are inherent problems in evaluating estimation error that resist easy solutions, but there also exists the potential for better estimation error measurement and analysis processes. In [14], we summarize our own and other relevant work, and propose a process for improved estimation error measurement and analysis. This method is, we argue, applicable for both researchers and practitioners. We have empirically evaluated the process and found that it improved the usefulness and meaningfulness of the error analysis, and the estimation process improvement work.

Project management and software engineering textbooks instruct that projects provide minimum and maximum effort intervals which include the actual effort with, for example, a likelihood of 98 per cent. The problem, hardly ever mentioned in the project management literature, is that the method leads to much too narrow an interval and, as a consequence, insufficient contingency buffers for the projects and poor project plans. In several studies, we show a process built on the same idea that we use to improve the uncertainty assessment of estimation models as proposed in [55], can be used to improve the uncertainty assessment of judgment-based effort estimates [34, 50, 57]. The simple idea behind the process is to use the error distribution of previous projects as the main input for the prediction intervals of future projects. Our work on this realism-improving principle has attracted attention in both the psychology, and the management and forecasting literature [18, 68, 1, 71, 3, 73, 74, 62] and is also accepted as a new forecasting principle at the prime forecasting community publishing their principles at www.forecastingprinciples.com.

Most of the results of our studies have direct consequences on judgment-based estimation processes. We are in the process of summarizing these results as estimation principles in a textbook, but have also published parts of these results in practitioners' magazines [30] and presented them at several industrial conferences (see "Presentations" under publications on the Simula web-page[5]). These include guidelines on the removal of situational and human biases [39], better selection of estimators [38], the use of looking-back strategies to increase realism [50], a combination of estimates from different sources [52], the use of checklists to increase consistency [51], and better provision of training opportunities [31].

Both the software developing organizations and the clients frequently suffer from bids based on overoptimistic effort estimates, for example, financial losses, chaotic projects, low product quality, and loss of market opportunities [37, 28]. Based on available evidence we identify several of the process elements leading to overoptimistic bids; examples are: Invitation of many bidders; focus on price in the selection of providers; lack of resources to thoroughly evaluate provider competence; information about price expectations included in the estimation material; description of the project as "small"; and bidding material with information about future opportunities. Particularly interesting, since it provides input to the understanding of the mental process of judgment-based effort estimation, is that the following bidding process led to 40 per cent lower bids compared with a one-step bidding process: Request of a specification containing more requirements than were actually needed; and request for bid updates based on a reduced set of requirements [37, 28]. Both studies took place in controlled experiments in field settings and have, we believe, a high external validity. We summarize our results as guidelines for software practitioners and clients in [25].

Improvement of Research Methods

Our typical research process, following the selection of a research topic, has been as follows:

1. Exploratory studies in the field, for example, surveys documenting the overoptimism, overconfidence bias.
2. Proposal of explanations (theories) on the reasons for these effects, based on previous research from various domains.
3. Laboratory experiments with a focus on validity of the explanations in software engineering contexts.
4. Field experiments on the robustness and relevance of the explanations.
5. Proposal of changes in estimation processes.
6. Laboratory and field experiments on the effect of the proposed changes in estimation process.

Our recommendations on when to use different research methods are, to some extent, described in our paper [20]. The opportunities provided at Simula Research Laboratory, namely, the freedom in the use of large-scale resources, coupled with

[5] www.simula.no

the opportunity to conduct realistic experiments and work in an environment that stimulates innovation in empirical software engineering methods, have been essential for the described improvements of research methods.

The challenges met when studying judgment-based effort estimation have led us to explore and advance several research methods previously not applied in software engineering research contexts. The main research method innovations, from a software engineering research point of view, are:

1. The first randomized, controlled software engineering experiments in the field.
2. The first use of multicountry populations (outsourcing companies) and studies of regional differences, in controlled software engineering experiments.
3. Extensive use of laboratory experiments to transfer research results to software practitioners and to provide input for our own research at seminars for software practitioners.

In many domains, findings from research based on randomized, controlled experiments is considered to be the most reliable form of scientific evidence [60]. As an illustration, all new medicines and surgical procedures must undergo such trials before being approved. As the name suggests, randomized controlled experiments involve a random allocation of an intervention, for example, a change of estimation practice, to subjects and mechanisms to ensure that the treatment is the cause of the effect. While laboratory studies are useful for many purposes, that is, to document the *existence* of an effect in a controlled environment, controlled field studies are needed to study the *size* of an effect in realistic environments. As documented in, for example, [41], large estimation effects observed in laboratory settings are in some cases much lower, and not of importance, in real-life settings. Our use of randomized controlled experiments in field settings typically includes the following elements:

- *Design of the study.* The design includes the formulation of a precise and relevant research question. In our case, this research question is typically derived from proposed models of judgment-based effort estimation or from the need to test changed or new estimation methods. The type of data that needs to be collected is determined from the need to test hypotheses related to the research question and to understand the findings.
- *Natural settings.* It is frequently essential that the participating software developers behave as similarly to their normal work behaviour as possible in our studies. This is typically made possible by paying software consultants customary fees for the estimation work. When this is the case, it is possible to create a situation in which the companies are not participating in an experiment, but rather are completing routine paid work for a client.
- *Participant selection.* There are several biases that might ensue as a result of sending invitations to a large set of companies and using only those that offer a positive response as participants in our studies. For example, there might be a bias towards more low-skilled than high-skilled companies responding on the type of estimation tasks we are requesting. To ensure that the estimation skill is acceptable and representative for the type of developer skill we want to study, we typi-

cally request curriculum vitas from the software professionals in charge of the estimation work and allow only those who possess an acceptable level of expertise for the purpose of the study to participate. A multicountry population of companies and software developers is sometimes selected to enable more robust, less cultural-dependent results.

- *Randomized treatment.* The estimation material and instructions about estimation processes to be followed are typically presented in more than one version. This enables the testing of previously defined research questions, for example, a research question related to how a change in the presentation of information or a changed estimation process affect the effort estimates. Typically, a study has a control group representing the default estimation process and one or more treatment groups. The control group will typically receive non-manipulated estimation material and no estimation instruction or training. The allocation of software professionals and companies to a control group or a treatment group will be random.
- *The estimation context.* To ensure an optimal level of realism, the companies should be asked to both estimate and develop the software work. However, this would make this kind of study extremely costly, and we therefore typically either 1) ask only for the estimation work, or 2) let the estimation work be part of a bidding round in which only one (or a few) of the companies are asked to develop the software. We have experience with both alternatives and find that they represent two different estimation contexts, both of which are realistic and likely to lead to valuable research results.
- *The estimation task.* The software projects to be estimated should be selected to fit the purpose of the study. If, for example, purpose is not related to specification size, then there is no need to specify a very large software system.
- *Monitoring and data collection.* We have experienced that written communication (mainly email) is an efficient way to monitor and document the communication during the estimation work. When communicating by email, one researcher can handle the communication that is necessary when acting as a client of at least 20 software development projects in parallel, provided that he/she has good support from administrative personnel for contractual and financial purposes. It has been essential for the quality of the results that we have had highly qualified personnel to manage and evaluate the deliveries of the software companies.

We are, as far as we know, the first to use outsourcing companies from a variety of countries in both field and laboratory experiments. This includes companies from India, Pakistan, Nepal, Vietnam, Russia, Ukraine, Poland, Romania, Slovakia, Bulgaria, Belarus, Moldovia, and Serbia. The use of outsourcing companies has, at least, three advantages: It reduces the costs compared with the use of, for example, Norwegian software companies; the results become more robust compared with single-culture studies; and it enables the study of regional (or even cultural) differences. As an illustration, in an ongoing study, we seem to find that information deemed as intentionally misleading had a much larger impact on software developers in India and Pakistan compared with software developers in Eastern Europe. This suggests that there may be a cultural component related to the perception of irrelevant information that needs to be better understood and/or more fully addressed.

Our laboratory experiments are typically conducted at seminars for software practitioners. A typical laboratory experiment is as follows:

- Formulation of a research question that can be addressed by designing a study in which the software professionals solve small estimation tasks. Preferably, the research topic should be related to the topic at the seminar.
- Design of the experiment. The experiment typically involves a control group and one or more treatment groups.
- Preparation of estimation tasks meaningful to complete in, for example, 5–15 minutes time. This may restrict the estimation tasks to be based on the development of very small programs or the provision of rough estimates on larger tasks.
- Random division of the seminar participants into groups, whereby participants in different groups get different treatments.
- Motivation of the participants to conduct professional estimation work through the information that: a) This is input for our research, b) They may learn from it, and c) They will receive the main results during the seminar.
- Completion of the tasks by the seminar participants. Continuation of the seminar, while another researcher inputs the results in a spreadsheet with analysis support.
- Information regarding the outcome of the experiment's main results provided to the participants.

We have, almost without exception, received positive responses from the participants on this kind of participation in seminars.

26.3 Transfer to and Impact on the Software Industry

The BEST-project members have emphasized the transfer of relevant research results to the software industry in and outside Norway. Examples of how we have transferred the results are:

- Inclusion of our research results in the most recent and most popular estimation textbooks and courses on software engineering, for example, the textbook *Agile Estimating and Planning*, by Mike Cohn; the textbook *Software Estimation*, by Steve McConnell; and several international courses leading to the *Scrum Master* (agile development) title.
- Frequent publication of research summary results in practitioners' magazines, such as *IEEE Software*, for example, guidelines on judgment-based effort estimation [30], how to avoid impact on estimates from misleading and irrelevant information [39], how to avoid making or accepting bids based on overoptimistic effort estimates [25], a debate with Barry Boehm (the most prominent estimation model builder and researcher) on the strengths and weaknesses of model-based and judgment-based effort estimation [36], and regular presentation of our own and other's research results relevant to the software industry in a regular (every sixth week) column in Computerworld Norway[6]. Presentation of results in 10–20

[6] www.idg.no/computerworld/

practitioners' seminars and conferences per year (the presentation can be downloaded from "Presentations" under publications at the Simula web-page[7]). This includes the organization of our own annual estimation research seminar with key decision makers in the Norwegian public and private sector.

- Spread of our research results on web-resources. This includes writing an article on software development effort estimation on Wikipedia[8], establishing a user group on software cost estimation at www.forecastingprinciples.com, offering easy download of our research results with practical implications for the software industry at www.simula.no/research/engineering/projects/best/bibliography, and offering free access, to the world, the most comprehensive database of studies on effort estimation at www.simula.no/BESTweb.
- Different types of collaborations with Norwegian IT-companies. This includes advisory work, company-internal seminars, and more formal collaborations in which a PhD student is supported with grants from an industry partner to implement estimation processes based on our research and to conduct valuable estimation research inside the organization.
- Building of two software companies based on the estimation research.

The total effect of our strong emphasis on the transfer of research results to and impact on the software industry is difficult to assess. However, based on numerous emails from, and personal communication with, software developers worldwide, we believe that many companies have changed their approach to estimation work based on our research results. The spread of certain research results is likely to be slow, and we expect that the main impact of our process change proposals is yet to come. In particular, we expect that our results on improved uncertainty assessment processes will be: implemented in more and more organizations, included as textbooks are updated, and incorporated into project management and effort estimation efforts throughout the world.

26.4 Conclusions and Future Work

The BEST-project proposes several estimation methods, principles, and guidelines with documented positive effects on estimation accuracy. The project has not yet been able to achieve its goal of developing a comprehensive model of the mental process involved in judgment-based effort estimation. It has, however, achieved a sound basis for the achievement of this goal and for the use of such a model to improve the effort estimation accuracy of the software industry. The project has the required multidisciplinary competencies (with researchers from psychology, education, software engineering, economics, and forecasting), it has acquired the knowledge base through extensive previous work on the topic, and it is infused with the professional background necessary to understand the practical implications of the research findings in software development contexts (namely, many of the project members are,

[7] www.simula.no

[8] en.wikipedia.org/wiki/Software_development_effort_estimation

or recently were, software practitioners). The project plans to continue its parallel focus on building a model of the mental steps involved in judgment-based effort estimation and transferring the resulting insight into practical estimation processes. The project also plans to initiate several commercial innovations based on recent and future research findings and has launched a new software company for this purpose.

Acknowledgments Current and former members of the BEST-project: Kristin Børte, Stein Grimstad (project leader 2008–2009), Tanja Miljana Gruschke, Nils Christian Haugen, Magne Jørgensen (project leader 2001–2008), Alf Børre Kanten, Kjetil Johan Moløkken-Østvold and Morten Onshuus. Others who have made research contributions to the project: Scott Armstrong, Bente Anda, Tore Dybå, Aiko Fallas Yamashita, Bjørn Faugli, Torleif Halkjelsvik, Jo Hannay, Ulf Indahl, Geir Kirkebøen, Barbara Kitchenham, Martin Shepperd, Karl Halvor Teigen and Dag Sjøberg.

References

[1] G. de Venter and D. Michayluk. An insight into overconfidence in the forecasting abilities of financial advisors. *Australian Journal of Management*, 32(3):545–557, Mar 2008.

[2] J. S. Armstrong. Principles of forecasting: A handbook for researchers and practitioners, 2001.

[3] D. V. Budescu and N. Du. Coherence and consistency of investors' probability judgments. *Management Science*, 53(11):1731–1744, Nov 2007.

[4] C. H. Castore and J. C. Roberts. Subjective estimates of own relative riskiness and risk taking following a group discussion. *Organizational Behaviour and Human Performance*, 7(1):107–120, Feb. 1972.

[5] R. M. Dawes, D. Faust, and P. E. Meehl. Clinical versus actuarial judgment. *Science*, 243:1668–1674, 1989.

[6] T. DeMarco. *Controlling software projects*. Yourdon Press, 1982.

[7] J. J. Dolado. On the problem of the software cost function. *Information and Software Technology*, 43(1):61–72, Jan. 2001.

[8] T. Dybå, B. Kitchenham, and M. Jørgensen. Evidence-based software engineering for practitioners. *IEEE Software*, 22(1):58–65, 2005.

[9] J. S. Edwards and T. T. Moores. A conflict between the use of estimating and planning tools in the management of information systems. *European Journal of Information Systems*, 3(2):139–147, 1994.

[10] S. Epstein. *Cognitive-experiental self-theory: An integrative theory of personality*. The relational self: Theoretical convergences in psychoanalysis and social psychology. Guilford Press, 1991.

[11] L. Farr. Factors that affect the cost of computer programming v.1. Technical report, united states air force Electronic Systems Division, L.G. Hanscom Field, Bedford, Massachusetts, jul 1964.

[12] P. Goodwin. *Enhancing judgmental sales forecasting: The role of laboratory research. Forecasting with judgment*, pages 91–112. John Wiley & Sons, 1998.

[13] P. Goodwin. Improving the voluntary integration of statistical forecasts and judgment. *InternationalJournal of Forecasting*, 16(1):85–99, Jan - March 2000.

[14] S. Grimstad and M. Jørgensen. A framework for the analysis of software cost estimation accuracy. *Proceedings of ISESE*, pages 58–65, 2006.

[15] S. Grimstad and M. Jørgensen. Inconsistency in expert judgment-based estimates of software development effort. *Journal of Systems and Software*, 80(11):1770–1777, 2007.

[16] S. Grimstad, M. Jørgensen, and K. Moløkken-Østvold. Software effort estimation terminology: The tower of babel. *Information and Software Technology*, 48(4):302–310, 2006.

[17] W. M. Grove, D. H. Zald, B. S. Lebow, B. E. Snitz, and C. Nelson. Clinical versus mechanical prediction: A meta-analysis. *Psychological assessment*, 12(1):18–30, 2000.

[18] K. HalvorTeigen. More than x is a lot: Pragmatic implicatures of one-sided uncertainty intervals. *Social Cognition*, 26(4):379–400, Aug 2008.

[19] K. R. Hammond, R. M. Hamm, J. Grassia, and T. Pearson. Direct comparison of the efficacy of intuitive and analytical cognition in expert judgment. *IEEE Transactions on systems, man, and cybernetics*, 17(5):753–770, Sep - Oct 1987.

[20] J. Hannay and M. Jørgensen. The role of deliberate artificial design elements in software engineering experiments. *to appear in IEEE Transactions on Software Engineering*, 2008.

[21] F. J. Heemstra and R. J. Kusters. Function point analysis: Evaluation of a software cost estimation model. *EuropeanJournal of Information Systems*, 1(4):223–237, Dec. 1991.

[22] J. Hihn and H. Habib-Agahi. Cost estimation of software intensive projects: A survey of current practices. *Proceedings of International Conference on Software Engineering*, pages 276–287, 13-16 May 1991.

[23] M. Jørgensen. A critique of how we measure and interpret the accuracy of software development effort estimation. *Proceedings of 1st International Workshop on Software Productivity Analysis and Cost Estimation*, pages 15–22, 2007.

[24] M. Jørgensen. A preliminary model of judgment-based project software effort predictions. *Proceedings of IRNOP VIII*, pages 661–668, 2006.

[25] M. Jørgensen. How to avoid selecting providers with bids based on overoptimistic cost estimates. *To appear in IEEE Software*, May/June, 2009.

[26] M. Jørgensen. Selection of effort estimation strategies. *submitted to International Journal of Forecasting*, 2008.

[27] M. Jørgensen. Estimation of software development work effort: Evidence on expert judgment and formal models. *International Journal of Forecasting*, 23(3):449–462, 2007.

[28] M. Jørgensen. The effects of the format of software project bidding processes. *InternationalJournal of Project Management*, 24(6):522–528, 2006.

[29] M. Jørgensen. The "magic step" of judgment-based software effort estimation. *Proceedings of International Conference on Cognitive Economics*, pages 105–114, 2005.

[30] M. Jørgensen. Practical guidelines for expert-judgment-based software effort estimation. *IEEE Software,* 22(3):57–63, 2005.

[31] M. Jørgensen. A review of studies on expert estimation of software development effort. *Journal of Systems and Software,* 70(1-2):37–60, feb 2004.

[32] M. Jørgensen. Regression models of software development effort estimation accuracy and bias. *Empirical Software Engineering,* 9(4):297–314, 2004.

[33] M. Jørgensen. Top-down and bottom-up expert estimation of software development effort. *Information and Software Technology,* 46(1):3–16, 2004.

[34] M. Jørgensen. Realism in assessment of effort estimation uncertainty: it matters how you ask. *Software Engineering, IEEE Transactions on,* 30(4):209–217, 2004.

[35] M. Jørgensen. How much does a vacation cost? or what is a software cost estimate? *Software Engineering Notes,* 28(6):5–5, 2003.

[36] M. Jørgensen and B. Boehm. Software development effort estimation: Formal models or expert judgment? *IEEE Software,* (Jan/Feb), 2009.

[37] M. Jørgensen and G. Carelius. An empirical study of software project bidding. *IEEE Transactions on Software Engineering,* 30(12):953–969, 2004.

[38] M. Jørgensen, B. Faugli, and T. M. Gruschke. Characteristics of software engineers with optimistic predictions. *Journal of Systems and Software,* 80(9):1472–1482, 2007.

[39] M. Jørgensen and S. Grimstad. Avoiding irrelevant and misleading information when estimating development effort. *IEEE Software,* May/June:78–83, 2008.

[40] M. Jørgensen and S. Grimstad. Judgment-updating among software professionals. *Proceedings of The 2nd international conference on software knowledge information management and applications (SKIMA),* pages 62–67, 2008.

[41] M. Jørgensen and S. Grimstad. The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. *submitted to IEEE Transactions on Software Engineering,* 2008.

[42] M. Jørgensen and S. Grimstad. Over-optimism in software development projects: “the winner’s curse”. *Proceedings of IEEE CONIELECOMP,* pages 280–285, 2005.

[43] M. Jørgensen and T. M. Gruschke. The impact of lessons-learned sessions on effort estimation and uncertainty assessments. *To appear in IEEE Transactions on Software Engineering,* 2008.

[44] M. Jørgensen and T. M. Gruschke. Industrial use of formal software cost estimation models: Expert estimation in disguise? *Proceedings of Empirical Assessment of Software Engineering (EASE),* pages 1–7, 2005.

[45] M. Jørgensen and T. Halkjelsvik. The effects of request formats on judgment-based effort estimation. *Submitted to Journal of Systems and Software,* 2008.

[46] M. Jørgensen, U. Indahl, and D. I. K. Sjøberg. Software effort estimation by analogy and "regression toward the mean". *Journal of Systems and Software,* 68(3):253–262, des 2003.

[47] M. Jørgensen, B. Kitchenham, and T. Dybå. Teaching evidence-based software engineering to university students. *Proceedings of 11th IEEE International Software Metrics Symposium,* pages 19–22, 2005.

[48] M. Jørgensen and K. Moløkken-Østvold. How large are software cost overruns? critical comments on the standish group's chaos reports. *Information and Software Technology*, 48(4):297–301, 2006.

[49] M. Jørgensen and K. Moløkken-Østvold. Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method. *IEEE Transactions on Software Engineering*, 30(12):993–1007, 2004.

[50] M. Jørgensen and K. Moløkken-Østvold. Eliminating over-confidence in software development effort estimates. *Proceedings of Conference on Product Focused Software Process Improvement*, pages 174–184, 2004.

[51] M. Jørgensen and K. Moløkken-Østvold. A preliminary checklist for software cost management. *Proceedings of IEEE International Conference on Quality Software*, pages 134–140, 2003.

[52] M. Jørgensen and K. Moløkken-Østvold. Combination of software development effort prediction intervals: Why, when and how? *Proceedings of IEEE Conference on Software Engineering and Knowledge Engineering*, pages 425–428, 2002.

[53] M. Jørgensen and M. Shepperd. A systematic review of software cost estimation studies. *IEEE Transactions on Software Engineering*, 33(1):33–53, 2007.

[54] M. Jørgensen and D. I. K. Sjøberg. The impact of customer expectation on software development effort estimates. *International Journal of Project Management*, 22:317–325, 2004.

[55] M. Jørgensen and D. I. K. Sjøberg. An effort prediction interval approach based on the empirical distribution of previous estimation accuracy. *Information and Software Technology*, 45(3):123–136, mar 2003.

[56] M. Jørgensen and D. I. K. Sjøberg. Impact of effort estimates on software project work. *Information and Software Technology*, 43(15):939–948, 23 Dec. 2001.

[57] M. Jørgensen and K. H. Teigen. Uncertainty intervals versus interval uncertainty: An alternative method for eliciting effort prediction intervals in software development projects. *Proceedings of International Conference on Project Management (ProMAC)*, pages 343–352, 2002.

[58] M. Jørgensen, K. H. Teigen, and K. Moløkken. Better sure than safe? overconfidence in judgement based software development effort prediction intervals. *Journal of Systems and Software*, 70(1-2):79–93, feb 2004.

[59] B. Kitchenham, T. Dybå, and M. Jørgensen. Evidence-based software engineering. *Proceedings of International Conference on Software Engineering (ICSE'04)*, pages 273–281, 2004.

[60] J. M. Lachin, J. P. Matts, and L. J. Wei. Randomization in clinical trials: Conclusions and recommendations. *Controlled Clinical Trials*, 9(4):365–374, 1988.

[61] M. Lawrence, P. Goodwin, M. O'Connor, and D. Onkal. Judgmental forecasting: A review of progress over the last 25 years. *International Journal of Forecasting*, 22(3):493–518, 2006.

[62] S. J. Mason, J. S. Galpin, L. Goddard, N. E. Graham, and B. Rajartnam. Conditional exceedance probabilities. *Monthly Weather Review*, 135(2):363–372, Feb 2007.

[63] K. Moløkken and M. Jørgensen. Expert estimation of web-development projects: Are software professionals in technical roles more optimistic than those in non-technical roles? *Empirical Software Engineering,* 10(1):7–30, 2005.

[64] K. Moløkken and M. Jørgensen. A review of software surveys on software effort estimation. *Proceedings of International Symposium on Empirical Software Engineering,* pages 223–230, 2003.

[65] K. Moløkken-Østvold, N. C. Haugen, and H. C. Benestad. Using planning poker for combining expert estimates in software projects. *To appear in Journal of Systems and Software,* 2008.

[66] K. Moløkken-Østvold and M. Jørgensen. A comparison of software project overruns – flexible vs. sequential development models. *IEEE Transactions on Software Engineering,* 31(9):754–766, 2005.

[67] K. Moløkken-Østvold and M. Jørgensen. Group processes in software effort estimation. *Empirical Software Engineering,* 9(4):315–334, 2004.

[68] D. A. Moore and P. J. Healy. The trouble with overconfidence. *Psychological review,* 115(2):502–517, Apr 2008.

[69] T. T. Moores and J. S. Edwards. Could large uk corporations and computing companies use software cost estimating tools? - a survey. *European Journal of Information Systems,* 1(5):311–319, 1992.

[70] T. Mussweiler. Comparison processes in social judgment: Mechanisms and consequences. *Psychological review,* 110(3):472–489, 2003.

[71] G. Overskeid. They should have thought about the consequences: The crisis of cognitivism and a second chance for behavior analysis. *Psychological Record,* 58(1):131–151, Win 2008.

[72] D. E. Sanders and L. P. Ritzman. On knowing when to switch from quantitative to judgemental forecasts. *International Journal of Forecasting,* 11(6):27–37, 1991.

[73] K. H. Teigen, A. M. Halberg, and K. I. Fostervold. More than, less than, or minimum, maximum: How upper and lower bounds determine subjective interval estimates. *Journal of Behavioral Decision Making,* 20(2):179–201, Apr 2007.

[74] K. H. Teigen, A. M. Halberg, and K. I. Fostervold. Single-limit interval estimates as reference points. *Applied Cognitive Psychology,* 21(3):383–406, Apr 2007.

[75] Y. Trope and A. Liberman. *Social hypothesis testing: Cognitive and motivational factors. Social psychology: Handbook of basic principles,* pages 239–270. Guilford Press, 1996.

[76] R. G. Webby and M. J. O'Connor. Judgemental and statistical time series forecasting: A review of the literature. *International Journal of Forecasting,* 12(1):91–118, March 1996.

27

FAULTY UNTIL PROVED CORRECT

An interview with Lionel Briand by Dana Mackenzie

On 4 June 1996, one of the most expensive disasters in the short history of software engineering took place.

It was the day of the maiden launch of Ariane-5, the European Space Agency's newest rocket. Designed to be able to lift six metric tons of payload into geosynchronous orbit, Ariane-5 had taken ten years to develop, at an estimated cost of seven billion dollars. But on that day, the flagship of the ESA turned into an expensive fireworks display. Thirty-nine seconds into flight, Ariane veered sharply off-course and had to be blown up by an automated self-destruct program.

Within two months, an official inquiry had determined the cause of the problem: a mistaken line of software. Specifically, the rocket's inertial reference system had been programmed to convert a variable called Horizontal Bias from a 64-bit floating point number into a 16-bit integer. Unfortunately, the number had grown too large to convert. The same line of code had been present in Ariane-5's predecessor, Ariane-4, but in that smaller rocket the variable in question (a measure of horizontal movement) had never gotten large enough to trigger an error.

After the initial error, a cascade of other failures occurred, which were all preventable. Perhaps most glaring, the commission found, was that the inertial reference systems were programmed to turn off in case of a failure–in effect, turning a recoverable error into a catastrophic one.

"The reason behind this drastic action lies in the culture within the Ariane programme of only addressing random hardware failures," the commission wrote. "The exception which occurred was not due to random failure but a design error. The exception was detected, but inappropriately handled because the view had been taken that software should be considered correct until it is shown to be at fault. The Board has reason to believe that this view is also accepted in other areas of Ariane-5 soft-

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_27,

ware design. The Board is in favor of the opposite view, that software should be assumed to be faulty until applying the currently accepted best practice methods can demonstrate that it is correct."

In other words, even though only one line of code had malfunctioned, the entire attitude of the Ariane engineers toward software needed re-thinking. One of the quality assurance experts that ESA hired to change the attitude was Lionel Briand, now a member of the Software Engineering department at Simula. "It was quite a tense experience–very interesting, but not so pleasant," he says. Some large and powerful contractors would be very unhappy with him if he criticized their procedures. But on the other hand, if a software problem caused yet another failure of the rocket, the consequences would be even worse. Now, of course, he is able to relax and laugh about it. "Fortunately, not only the second launch but all the later launches have gone off without a hitch!" he says.

Ariane's cautionary tale could potentially be repeated many times over the coming years, as more and more engineered systems–in cars, ships, factories, banks, and elsewhere–become dependent for their most critical functions on software. "Software is an engineering artifact," Briand is fond of saying. And yet we persist (and even more worrisome, engineers persist) in regarding it as something different. Perhaps because software is abstract and conceptual, engineers do not see it as something that requires testing in just the same way as any other engineering artifact.

However, thanks to Briand's work and his forceful personality, the new concept of software–faulty until proven correct–is beginning to take hold in Norway and elsewhere. From 1999 to 2008, as a professor at Carleton University in Ottawa, Canada, Briand worked on the subject of software verification and validation. He advocated an approach called model-based verification, in which the software is both developed and verified by using modeling languages, both textual and graphical, that have precise semantics and can be used to carefully model the expected behavior of software systems. But from his academic chair, even though he was one of the most highly-cited experts in the world, Briand felt that he was not doing enough to translate software verification and validation into daily practice. In 2007 he accepted an offer to come to Simula, in part because it would provide him the opportunity to affect industrial practice directly. In less than two years, he has attracted several new partners to Simula, including for example the Swedish manufacturing giant ABB; Norwegian shipbuilder Kongsberg; and risk-management firm DNV.

When we interviewed Briand for this book, we were told to expect an interesting and long conversation, and Briand did not disappoint us. With his infectious laugh and his never-flagging enthusiasm, he is always willing to talk. Here are some excerpts from our interview.

"Let's start with your personal background. What did you do before you came to Simula and why did you decide to come here?"

"Before I moved to Norway I lived in four countries. I was born and studied in France, and then I went to finish my PhD work and do a postdoc in the U.S. I had a short stay in Montreal for 13 months, but then I got a very good offer to manage a department

Lionel Briand

focused on software quality at one of the Fraunhofer Research Institutes in Germany. Then, eventually, after four years, because I wanted to live in Canada and become Canadian, I moved back to Canada and became a professor.

"As a professor in Canada, every six years you have a sabbatical. I came to spend my sabbatical here, and during that time I got to know Norway and Simula. I already had a collaboration with them even before I came for a sabbatical, mostly with Erik Arisholm. Still, it was when I came here on sabbatical that I really came to know the institute, the country, and the city. So I went back to Canada and Simula asked me if I was interested to come back. After a period of reflection I decided to come back here for good."

"That sounds like a pretty major change."

"It was a major change, and it was also a big risk. I had a professorship, a Canada research chair, which is the highest academic award in Canada. So I had to give up a lot. It was a very hard decision. I had to give up a lot of things that would be very difficult to get back if I wanted to. But my outlook on life is that you should go where your heart leads you. You cannot just take an accounting approach to your life decisions: how much do I have, how much can I lose?"

"Why did Simula fit in with your goals?"

"There were two things: I liked Simula and I liked Oslo. There was both a personal and a professional side. First, Simula as an institution is a unique concept, because it combines substantial basic funding for research with a corporate-like management style where clear decisions are made to focus on certain research areas, and decisions are made quickly, to optimize the way research funding is used. There is also a project-based organization that facilitates collaboration with industry. It's not just a professor and a bunch of students. There are real projects with real teams which allow you to tackle larger problems and work more closely with industry. But at the same time there is basic funding that allows you freedom to take more risks, and to explore more fundamental issues.

"One thing I wanted to do was more industry-driven research. In my field, too much of the research has negligible impact on practice. Of course, this is a very personal opinion.

"The field I work on, software engineering, is by nature an engineering field. But historically, it is kind of an offspring of computer science. So the paradigm of research is not the same as in all engineering disciplines, where people define real problems in collaboration with industry or hospitals, for example, and try to find scalable solutions. Computer science research has a different attitude, more based on self-identified problems, or attacking very narrow sub-sub-problems of larger problems, without paying much attention to scalability in practice.

"The other reason I came here, as I said, was the city. I'm Canadian, and I've been in Canada a long time. But sometimes I still wanted to come back to Europe. Oslo is an interesting European capital, with snow in winter, and there's space here, as there is in Canada. It's one of the few European countries where you have space."

"You are actually certified in Canada as an engineer. Is that unusual for a software developer?"

"Yes, because again it's a relatively new discipline. You find a lot of people in industry who develop software and do not have any formal education in software engineering. You cannot call yourself a civil engineer or a mechanical engineer if you do not have the right degree. But somehow in software engineering it's possible. That's what they are trying to change in Canada. In software engineering worldwide, it's not yet a widespread practice. People still call themselves software engineers even though they are not a certified engineer. But I don't think there is any question that this has to change, because software systems are engineering products; they are engineering artifacts."

"Can you elaborate on why software development is an engineering discipline?"

"Software is an engineering artifact because it is technically complex and plays a critical role in all aspects of life and industry. Therefore it must be designed and verified according to rigorous, technical practices.

"We're in Norway, so let's talk about the energy and maritime sectors. If you look at how much software is running on oil platforms or ships, this is where most of the complexity resides. Not only that, but software is in charge of making many systems safe. If software doesn't work, what is at risk is the safety of people and infrastructures and the environment. In Norway, there are a number of papers reporting incidents where, for example, the steering control of ships was lost. Fortunately, due to luck, those incidents did not lead to loss of life or environmental damage. But it was just luck because the boat was not near to a dock or an offshore platform.

"Safety is one aspect but of course it is not the only one. Our whole economy and industry depend on software. A report in 2002 by the U.S. National Institute of Standards and Technology (NIST) showed that software failures were responsible for the direct loss of sixty billion dollars a year in the U.S. alone. A direct loss! Of course you also have indirect damages. You cannot quantify loss of reputation[1].

"My field of research is called software validation and verification, so safety is not my only interest. But it's an increasing interest because I am working with the energy and maritime sectors where safety is a major concern—with ABB, Kongsberg, and DNV, and those people are very concerned with safety. DNV is the Norwegian Veritas, which basically certifies systems in the maritime and energy sectors—ships and offshore platforms. It is very concerned with safety with respect to human losses and the environment."

[1] As a case in point, the Ariane-5 failure caused no injuries, but it took years for the European Space Agency to fully recover its and Ariane's reputations.

"What do verification and validation mean?"

"Verification is about finding faults in the system. Validation is about ensuring that the system is ready for delivery. So the goals are different. Our work is mostly focused on verification so far. Both are about making sure the system does what it is supposed to, but verification is focused on finding problems, while validation is about assessing that the system is dependable enough, whatever aspect of dependability you are interested in. In the telecom domain you are interested in robustness. In a safety critical domain you are interested in safety. In the space domain, when they send a satellite up and they want to validate the flight control software, they want to ensure that the likelihood of failure is very very low, for example less than 10^{-5}."

"So when you're validating the system, you will be using it in the most routine mode?"

"Exactly, when validating, yes. But when verifying, you're trying to find out how you're likely to make the system fail or where you're likely to find problems in the design.

"In fact, designing verification techniques is a multidisciplinary thing. We use all kinds of technologies. We use modeling technologies, search techniques that come from artificial intelligence, such as evolutionary computing. The main challenge of software verification is that exhaustive verification is impossible[2]. If exhaustive verification is impossible, how do you get a sufficient level of confidence? That's a pretty tough dilemma. But you cannot ignore it, because those systems have to be verified."

"I understand that you have just gotten a contract with DNV or Veritas. Can you talk about what that contract involves?"

"Veritas has very advanced expertise in many areas, because the systems they certify are composed of many components–mechanical components, electrical components, software, etc. When you certify a ship, you have to verify that everything is safe. They decided they needed to advance their expertise in software systems, because they realized–and they have been realizing it for a while–that the software component is increasingly important and increasingly complex. So they set up a project focused on software systems in the maritime domain. The point is to identify what are the most crucial problems to address in that domain–we have already started to do that–and investigate scalable solutions to address those problems. So it's a big, ambitious project. DNV is serious, and its investment is very substantial.

"This project is at a crossroads with something else I've been working on, called model driven development. In all engineering disciplines, people tackle complexity by abstraction, by building models. Models are used to analyze engineering artifacts before they are built. That's what people do! People have blueprints of electric circuits, of bridges before the bridge is even built. That's exactly what we need to do

[2] Modern software systems are so complex that it is impossible literally to check every state of the system, and every possible input to that state, and make sure the system responds correctly.

with software. We need to develop models of software early on so that we can try to analyze whether the software has the required properties based on its design."

"You mention scalability a lot. Is that a problem people weren't aware of at first?"

"I mention it a lot because software systems are becoming increasingly complex, but at the same time I find that there has been little attention in the research community to devising scalable solutions. Again, it's due to the scientific tradition of software engineering, being an offspring of computer science. In normal engineering disciplines, people focus on developing practical, scalable solutions, because they have to."

"Are the search techniques that you use for model-based verification more scalable than exhaustive verification would be?"

"That's the idea of using search technologies, for example from evolutionary computing, such as genetic algorithms. You cannot do exhaustive solutions, so you search for problems using heuristics. Genetic algorithms have been used in many other engineering disciplines. We are using them to search for problems in design and also to generate optimal test suites once the system is implemented, test suites that have a higher likelihood of finding problems. For example, in real-time systems, which have to respond to events within time constraints, we use genetic algorithms to generate test cases that stress the system to maximize the likelihood of missing deadlines. Also we use the same approach to find concurrency problems in concurrent software. That is, in software that is designed to have concurrent threads executing, which access common shared data, we have to be sure that the threads don't interfere. A typical problem is that the threads get deadlocked, because they are all waiting on each other to do something[3].

"We are going to use such technologies on the DNV project to find safety problems. To put it in an intuitive way, can a software system violate safety constraints? You want to find such problems early on, and once the system is developed you want to test for it. This is why we need models, representations that are analyzable. At the same time, once you have done your best and implemented the system, you still need to check that the actual system is safe. Ideally the model should be used for both."

"Can you tell me about the origin and the impact of the Unified Modeling Language (UML)?"

"UML became the standard for the first time, I think, in 1997. The problem we have had in software engineering for a long time is that, as for any other engineering discipline, to represent the specification of the design of an engineering artifact you need a modeling notation. In software engineering, for a long time we didn't have a standard notation that was agreed upon. There was a notation war. Every academic

[3] Again, genetic algorithms give Briand's group a high probability of detecting conditions that would lead to such a deadlock.

had a better idea than the other. Every consulting firm had its own tool and notation. It was a complete mess.

"At some point the Object Management Group, an industry consortium that includes companies like IBM, Microsoft and others, decided that we needed to put all the expertise together to create an industry-wide standard. And so they did.

"Of course this has been a huge revolution. All of a sudden, we had a modeling notation that was widely supported by many tools and many technologies that people could choose from. So you were not using one technology supported by a vendor who could disappear. There are increasingly a large number of commercial tools and even open-source tools that can be used by industry partners to support their own current development. This has been a huge revolution from a practical standpoint.

"Even better, that standard is extensible. That is, anybody can extend it to fit the specific needs of a domain. You can still use all the tools around, and the tools are designed to be able to extend them, to add functionalities. That's the case of one of the market-leading tools by IBM, with whom we are working a lot. They have developed a tool called Rational Software Architect. Here in Norway we have a privileged collaboration with IBM where we use their technologies and they provide us those technologies for free.

"Thus we have the advantage of a standard, a widespread choice of technologies we can use, and it's an extensible standard. Without it, it would not be possible to do the type of research we do with industry."

"Are the companies aware of this, or are you educating them as you go along?"

"They are partly aware of it, but we are helping them by increasing their awareness, let's put it that way. Very often they need help because those decisions of what to use, when and for what are very complex decisions, and often that's not their main job."

"They aren't a software company."

"Yet this is the main ambiguity! They're not software companies, and yet software is at the core of their business. I think there are many companies that are becoming software companies without realizing it, you know? What does that mean, to be a software company? You think that you are not a software company because you are not selling software. But the investment in your business is such that 80 per cent of your investment is in software systems. Are you, or aren't you, a software company? To me you *are* a software company, to a large extent, when this happens."

"What are some other advantages of model-based verification?"

"In software engineering, it's not unusual to see that half of the expenditures go into testing the software, rather than designing it. In safety-critical domains, the majority of expenses are in the verification–more by far than in designing and implementing the system."

"Is that because it's not being done efficiently?"

"Of course the expense will always be significant, but it can be improved a lot. That's where I think model-driven verification has an important role to play. We want to improve the quality, but we also want to reduce the cost. The two are linked anyway. When the cost is reduced, then there are more resources available to do verification and validation, therefore it results in better quality. You cannot distinguish the productivity from the quality. When you improve one, you improve the other.

"Another problem, when you are testing software, is the problem of determining when you are done. What happens in practice in many cases, although it's not politically correct to say so, is that you are done when time is up! Is that really the way you want to verify software? Can you imagine doing that with analyzing the properties of a bridge? After two days, do you say that time is up and build the bridge?

"DNV has very precise procedures for checking the stability of oil platforms, their structural safety and other aspects I don't know about yet. So why shouldn't we have precise criteria for software, especially when it is playing an increasingly important and critical role?

"You can tell that people are realizing this. There is a huge need in industry. That's why in one year and a half we have already established many combinations, more than I can even deal with, even though I was new to Norway and didn't know anybody and had very few contacts. We have collaborations with Tandberg, ABB, Telenor, DNV, and Kongsberg. Why is that? It's not because I have some magic trick. I don't. It's because there is a need. It's blatant. The technologies are just not there to cope with the scale of the problem.

"Of course, consulting companies will tell you otherwise, that they have a solution for everything. But that's what consultants are supposed to say. The fact of the matter is that it's not true. There is a lot of research that needs to be done. It cannot be done without collaboration with industry. Research in this domain without the collaboration of industry will not lead to usable, scalable solutions.

"Of course, when you do research with industry to devise innovative solutions, you have to understand the problems. There is a learning process from them to us in terms of a better understanding of the problems that we have to solve. And of course understanding the application domain in which we work. Before I came here, I worked in the aerospace and automotive sectors, so I didn't know anything about the energy and maritime sectors. I had no clue!"

"What are the biggest changes in shifting from aerospace to maritime?"

"I'm not sure yet, because I'm still learning! From what I've seen so far, there are a lot of commonalities. All the problems are the same but the priority is not exactly the same.

"For example, in the maritime energy sector, each offshore platform or ship involves so many suppliers. All that stuff has to integrate. This is a very critical problem in the maritime sector, a problem that is constantly raised. There is software in nearly every component. The software is often the glue between the pieces. Also, in Norway in particular, there is a very high concern for the environment, so that is a very important part of the safety problem.

“The environment is a problem in the automotive sector, too, but it’s not the same thing. The risk seems many times greater in the maritime sector, in terms of the damage that you can do to the environment with a ship or an offshore oil platform.”

28

SOFTWARE VERIFICATION — A SCALABLE, MODEL-DRIVEN, EMPIRICALLY GROUNDED APPROACH

Lionel C. Briand

Lionel C. Briand
Simula Research Laboratory

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_28,

PROJECT OVERVIEW

Software Verification and Validation

Research in software verification and validation (V&V) has been conducted for more than three decades. Software systems play an increasingly critical role in society and industry and their complexity tends to grow exponentially over time. As a result, existing V&V technology in many industry sectors is no match for the scale and complexity of software systems, and this makes it difficult to ensure the dependability of software systems in a cost-effective way. For example, safety-critical systems in various industries (e.g., aerospace, automotive, maritime, and energy) increasingly rely on software and need to be certified with respect to their safety. Despite international standards and practical guidelines, no cost-effective, well-established way to ensure software safety at a reasonable level exists.

Software V&V research develops algorithms, strategies, and tools to help develop and automate techniques to detect failures and correct faults in software systems. From a scientific standpoint, this research involves a variety of technologies that cover many technical domains, including software modeling, programming languages, static and dynamic analysis of source code, simulation, and optimization and search algorithms. These domains must be integrated to form a suitable, complete, and practical V&V solution.

Scientific Challenges

The main challenges regarding software V&V relate to devising solutions that scale up to the increasing complexity of software systems. In practice, resources to verify and validate software systems are limited, both in terms of available expertise and time. The challenges at a more detailed level are related to the effective automation of verification techniques and to the evaluation of their cost-effectiveness.

Most of the work on V&V performed at Simula takes a model-driven approach that relies on models of the behavior and properties of the designed software system. We make use of metaheuristic search techniques developed in evolutionary computing to reveal potential problems in the system design and to generate an optimal set of test cases with a high fault-revealing power. The focus of verification spans faults ranging from functional to safety, response time, and concurrency properties. Such an approach is markedly different from mainstream approaches, based for example on static analysis of source code or model checking, which, for different reasons, tend not to scale up to large systems.

Obtained and Expected Results

A number of model-driven techniques have been developed for class testing, component off-the-shelf testing, integration testing, and testing of non-functional properties such as response time, deadlocks, or starvation. Many other aspects of verification must be investigated, with a particular emphasis on safety and robustness. Moreover, providing automated solutions and investigating their cost and effectiveness on large systems requires attention.

We currently are in the process of establishing model-based testing practices at Tandberg in Oslo, and new projects with other companies are beginning. One of these projects, with Det Norske Veritas, that is of particular interest to Norwegian society, is the improvement of software V&V technology in the maritime and energy sectors, where complex software systems play an increasingly important role and where software failures can lead to dramatic consequences in terms of loss of human lives, damage to the environment, or significant financial losses.

SOFTWARE VERIFICATION — A SCALABLE, MODEL-DRIVEN, EMPIRICALLY GROUNDED APPROACH

28.1 Introduction

Software is present in most systems across all industries, including energy, automotive, health care, maritime, aerospace, and banking, to name just a few. Software systems are increasingly taking on safety- and business-critical roles and growing in complexity. One crucial aspect of software development is therefore to ensure the dependability of such systems, that is, their reliability, safety, and robustness. This is achieved by several complementary means of verification, ranging from early analysis of system specifications and designs to systematic testing of the executable software. Such verification activities are, however, difficult and time-consuming. This stems in part from the sheer complexity of most software systems and because they must accommodate changing requirements from many stakeholders.

Software verification potentially has a high impact on the dependability of systems and therefore their economical, human, and environmental impact. Exhaustive verification, however, even on smaller systems, is impossible to achieve and this often leads to the difficult dilemma of how to achieve sufficient confidence with limited resources. This chapter presents a personal assessment of the state of the art in software verification, its achievements and gaps, and a set of research directions that the author believes hold promise for the future.

28.2 Background

Fundamentals

The goal of software verification is to make software-based systems dependable. Software dependability is, however, a multipronged concept. Ideally, one would like software systems to be correct. It is, however, highly difficult to prove the correctness of even small programs [11]. The closest workable concept is that of *reliability*, which is defined as the probability that a system will perform its intended function during a specified period of time under stated conditions [17]. Moreover, there is more to dependability than just reliability. A system also needs to be robust and safe. A system is robust if it acts reasonably under severe, unusual, or illegal conditions [31].

Robustness is often associated with the concept of "graceful degradation", that is, the capacity of a system to provide partial functionality even under degraded conditions. Safety is related to whether a system can cause damage, for example, threaten human life or cause serious environmental damage. Those three aspects of dependability, namely, reliability, robustness, and safety, are related but distinct. They must all be addressed by verification with dedicated techniques.

There are three ways according to which one can influence dependability. First, through rigorous specification, design, and coding practices, one can limit the introduction of defects, although one cannot entirely avoid them. Second, one can attempt to detect defects as early as possible in the development life cycle, for example, by inspecting, analysing, or testing various artefacts, such as design documents, models, and source code. Such activities are referred to as *verification*. Last, software can be designed to be fault tolerant in order to contain run-time failures and, in the worst case, provide degraded but graceful functionality in the event of failure [38].

Verification can be performed according to different processes, depending on the artefact to be verified and the technology available. As far as non-executable artefacts are concerned, common practices include inspections, walkthroughs, and code reviews [46]. Those are all variants of informal but systematic manual or partially automated analyses of development artefacts, ranging from planning documents to specification and design documents to even source code. Alternatively, if the specification and design are represented as verifiable models (i.e., abstractions) bearing precise semantics, then automated model analysis can be considered. Model analysis is used here as a general term to refer to any form of analysis of model properties. For example, one may want to verify whether deadlocks are possible in a concurrent design [57]. Model analysis methods range from formal and exhaustive (e.g., model checking [31]) to search-based heuristics[1] [31]. Executable code can be tested, that is, executed in some controlled and systematic fashion in order to ensure that it properly implements its specifications. This chapter will focus on two types of verification: model analysis using search heuristics (simply referred to below as model analysis) and testing. The main reason is that, in the foreseeable future, these options are believed to be the only ones scalable to large systems.

Testing

Testing usually has a number of complementary objectives. It must, of course, be effective at triggering failures and therefore detecting faults. But it is also important that the testing process be automated and repeatable. It must be automated to be cost-effective, given the typical complexity of software systems, and it must be repeatable so that a precise understanding of the fault can be gained to verify that it was properly corrected. It must also be helpful in locating faults in the source code once a failure is observed, i.e., the fault localization problem [55]. Finally, it must be systematic in order to associate an expected level of dependability with a specific testing strategy.

[1] This form of analysis is not to be confused with model checking, which, as further discussed below, is an important approach to model analysis that has been getting increasing attention in recent years.

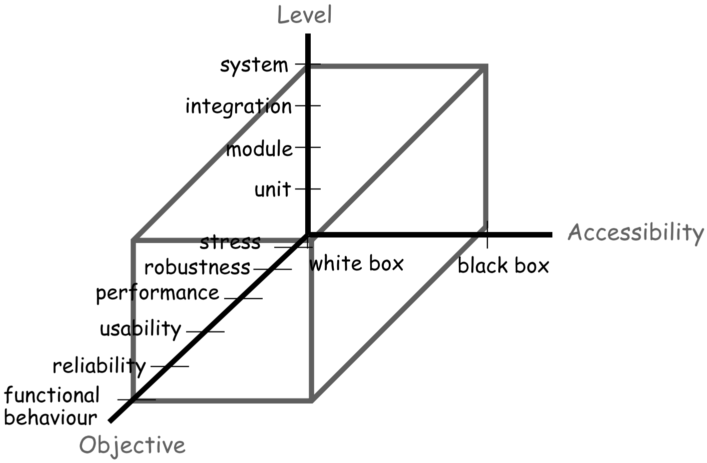


Figure 28.1 Dimensions of software testing.

There are many different testing activities in a typical software development process, as illustrated in figure 28.1. Such activities typically differ according to the type of faults they aim to find and the phase of verification during which they can be applied. Testing can focus on the verification of single units or components in isolation (i.e., unit or component testing). It can target the interactions of components or (sub)systems, which is usually referred to as integration testing, or it can encompass the testing of entire systems, whether on development or deployment platforms, which can be distinct and very different for embedded systems [57]. Test techniques are typically classified according to three main categories. Black-box techniques rely exclusively on some representation of the specification of the system under test (SUT): They do not use internal information regarding design or source code. White-box techniques rely on structural information obtained, for example, through source code analysis. Techniques that rely on partial information about the system's internal details, such as design documents, are often referred to as grey-box techniques. These different testing techniques are complementary, since they are used to target different types of faults during different verification activities [42]. One practical issue is to determine the appropriate combination of techniques in a given development context.

When testing is based on models of the system's behaviour or structure, the terms *model-based* and *model-driven* are typically used. The main idea, in the context of model-driven development (MDD) [51, 44], is to exploit the early models of the system's specifications and design to automate (part of) the test case generation process. The development models are transformed into test models that can then be exploited by specific test generation algorithms to satisfy certain coverage requirements [61]. The notions of test model and coverage criteria are illustrated in 28.2, where the test model could, for example, be a state machine model of the SUT and the coverage criterion could require that all transitions in the state machine be

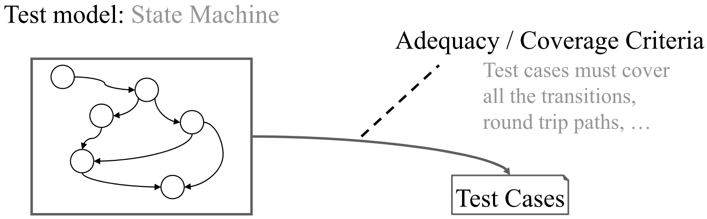


Figure 28.2 Deriving test cases from test models and coverage criteria.

covered by test suites. Based on a test model and an objective coverage criterion, mechanisms can be devised to automatically derive test cases, although this is not always an easy endeavour. The goal is then to check the conformance of a system implementation with a dedicated model representation, which is itself derived from specification and design information. Failures can be due to a fault either in the model or in the implementation. Models have also been shown to be useful in addressing the well-known oracle problem, which is the way the verdict of each test execution will be determined, since test case execution results can be compared against relevant information in the model.

Model Analysis

When the behaviour, structure, and other properties of the software are modeled during the specification and design stages, such models can be analysed to verify various relevant properties early in the development cycle, long before any testing can be performed. There are different ways to go about this, each differing in their level of formality, practicality, and scalability. One important field of research is model checking [31], which can be defined as "the process of checking whether a given structure is a model of a given logical formula". For example, the structure can be a finite state machine and the logical formula can be expressed in propositional logic or temporal logic. Such formulas may target safety, temporal, or concurrency properties. The advantage of the model-checking approach is that it systematically explores all reachable states (at a certain level of abstraction) of a system, thus providing strong confidence about whether certain properties hold. One of the challenges of model checking, however, is that on industrial-scale problems, it often faces a combinatorial explosion problem for which many solutions are still currently being investigated [31]. Another practical issue is that many model checkers require the use of modelling languages that are far away from current development practice and not always easy to use on large-scale problems, for example, temporal logic. The approach to model analysis focussed on in this chapter is aimed at being more scalable and is based on evolutionary or meta-heuristic search techniques [16]. In this situation, a fitness or objective function is defined to represent the violation of a property and a guided but random search technique is used to traverse the space of possible behaviours of the system being verified. The point is to uncover problems without having to explore the entire state space of the system. Such an approach does not guarantee that a property will never be violated but through the guided search it optimizes the chances of finding property violation occurrences if there are any. Furthermore, approaches based on search heuristics are usually not as demanding as model checking in terms of the level of formality required for the model input. This is also expected to facilitate the integration of model analysis with the other activities of software development. Examples will be discussed below.

State of the Art, State of Practice, and Problem Definition

The point here is not to provide a comprehensive overview of the state of the art in software testing. For this, the reader is referred to the several reviews on the subject [23, 49, 37]. What follows is, instead, a subjective assessment reflecting the author's experience trying to bridge research and practice and apply or tailor research results in practical situations.

Software-testing research has been a focus of attention for more than three decades, though the number of researchers in the field has grown exponentially in the last decade. This stems from the recognition that verification activities take up to 50 per cent of resources on typical development projects and far more in the context of safety-critical development. Furthermore, several recent papers have shown that current verification practices are far from satisfactory [14].

A large body of work exists on testing techniques based on control and data flow analysis of the source code [31]. Such white-box approaches have, however, shown practical limitations in terms of their scalability, e.g., when dealing with millions of lines of code, and automation. For example, the identification of infeasible control flow paths is generally an undecidable problem. It was suggested that in the context of testing larger components or systems, white-box testing could only be used to indicate what parts of the system lacked coverage once black-box testing was applied, thus leading to the refinement of black-box test suites [42]. As a result, in practice, white-box testing is used in its simplest form, such as statement coverage and, more rarely, edge/branch coverage. Very few tools [5, 35] go beyond these simplest strategies for code coverage, despite decades of research on the topic.

Another important body of work in the testing area concerns mutation analysis or fault-based testing [61]. The main idea is to define so-called mutation operators to automatically modify the SUT in small ways and generate large numbers of program mutants, i.e., programs with one incorrect change. Test suites are then refined until all mutants trigger at least one failure. The motivation is to expose weaknesses in the test data and ensure that all parts of the SUT are exercised during testing. The main problem with mutation testing is that typically many mutants are equivalent from a functional standpoint to the original program and identifying equivalent mutants is once again an undecidable problem. Second, even on small programs, very large numbers of mutants can be generated and this therefore requires valid sampling strategies to select a representative subset of mutants. Last, whether test suites that effectively detect mutants are also effective at detecting real faults is still in question, although recent evidence suggests that this is the case [33]. As a result, despite substantial research over the last two decades, to the knowledge of the author, only limited industrial application of fault-based testing has been reported.

Another area of intense research has been the use of state machine models to test communication protocols [49]. This research started three decades ago with the seminal article by Chow [59] on the W method. This research focussed, to a great extent, on finding strategies to traverse the finite state machines to guide testing and on automatically deriving distinguishing sequences of inputs to determine the resulting state of test sequences, thus helping determine whether the communication protocol conformed to its state machine specification. Though this body of work has impacted the verification of communication protocols, state-based testing in other areas is still a very rare practice. One reason is that very little is known regarding the cost-effectiveness of various test strategies based on state models [47]. Recent experiments [45] have shown that certain strategies are far from effective, even for small software components. Another reason is that the practice of state modelling in software development is still infrequent–with perhaps the exception of certain areas in embedded systems where it is required by standards–in part because it is not appropriate for every type of system but also because it is rather complex in large components or subsystems.

A large body of work addresses the problem of regression testing. The goal is to be able to minimise and prioritise regression test cases on every release [43, 36, 28]. This is important, since with the rise of incremental development, several releases of a software product are typically released every year. Since regression test suites that check whether unchanged functionality has not been broken by new changes can be quite large, re-running all regression test cases can be impractical or even impossible. This is where selection and prioritisation techniques come in. Most of these are based on source code control flow, data flow, and change analysis. Most existing studies are based on small, artificial programs and very little is known about the conditions under which such regression test techniques are beneficial. The gains in many cases seem to be rather small and test case selection often leads to faults remaining undetected. Though prioritisation is more promising, empirical results based on realistic conditions (i.e., real releases, changes, and systems) are rare. As a result, despite at least two decades of work on the subject, the results of academic research in regression testing are scarcely applied in practice and there is no comprehensive commercial tool supporting most techniques.

Other areas of intense research that have failed to transfer to industrial practice include testing based on logical specifications [19] and the use of combinatorial designs, especially in the context of testing software with many configuration parameters [60].

In recent years, with the maturation of the Unified Modeling Language (UML) [24, 50] as a standard modelling language for software, practical and model-based verification techniques have received increasing attention. Indeed, for the community working on model-based verification, UML has brought a number of advantages. It addressed a wide range of modelling requirements and came with an increasingly sophisticated set of tools and open-architecture technologies to help automate analysis and testing based on models. The UML is the modelling language used in the context of the Model Driven Architecture®(MDA®) standard supported by the Object

Management Group$^{\text{TM}}$(OMG$^{\text{TM}}$)[2], a large consortium of software industry leaders. Furthermore, the language can be used at different levels of formality and extended and tailored according to needs into so-called profiles. Also UML 2.0 contains many features that support the modelling of large-scale, complex systems. Though it can be used at different levels of rigour, UML has tried to bring together best practices from many world experts and past successful methods and a real attempt to make modelling practical and scalable has been made through the work around MDA. Recently, UML was even extended to define a UML testing profile [15] whose main goal was to provide a way to model testing information, e.g., test cases, suites, and harnesses, using also the UML and its associated tools, thus facilitating the development of test tools and the interchange of test data among them. Other profiles exist, for example, in the areas of real-time and embedded systems [39] and safety-critical software [38]. Such profiles, as standards approved by the OMG, can be used to develop models that are then appropriate as a source of information for automating verification, either through model analysis or model-based testing. The work on using UML models to automate testing is, however, still fragmentary and, as discussed in [32], most techniques are superficially defined and not automated and validated to an extent that makes them interesting solutions to consider.

At the same time that MDA was evolving as a standard, a new field of research emerged focussing on hard, long-standing test automation issues: evolutionary testing. The basic idea is to transform an automation problem into a search/optimisation problem. Evolutionary search techniques, also referred to as meta-heuristic search techniques, are used to solve typically hard search problems in large search spaces [16]. They have shown to be effective for a variety of testing problems, such as automating the generation of test suites to achieve code coverage [58]. More recently, these techniques have been used to address non-functional testing problems [52, 7] such as execution time deadlines and safety properties. Despite promising results, however, the scalability and effectiveness of evolutionary testing to address realistic test automation problems still remain to be investigated [40].

A recurring problem in past testing research is related to the scalability and cost-effectiveness of the proposed test techniques. How can one gain sufficient confidence in complex software systems when limited time and resources are available? There is obviously no perfect solution to this dilemma. As further developed below, however, there are reasons to believe that certain types of model-based analysis and testing strategies, supported by evolutionary search techniques, can be combined to address large-scale software verification in a cost-effective and scalable manner.

Although mostly an academic exercise at this point, model checking is the most common approach to model analysis and consists in verifying that a system or, rather, a model thereof, complies with a property, for example related to safety or concurrency, by exhaustively exploring all its reachable states [31]. It therefore requires that system models, for example state machines, and property models such as temporal logic formulas, be developed to be applicable. These models must comply with the specific notation used by the selected model checker (e.g., Promela in SPIN

[2] www.omg.org

[31]). Despite many claims, its practice, however, is still very rare and very little evidence exists to show that it can scale up to real verification problems and whether this is applicable in most software development contexts. The main problems stem from its underlying principle of exhaustively searching a state space and, though a number of approaches have been investigated to alleviate this problem, no generally satisfactory solution has been devised.

28.3 Requirements for Model-based Verification

The purpose of this section is to clearly identify the requirements to address the problems stated in the previous section. This will help us structure our discussion in the next section and provide clearer arguments to support the proposed research directions.

Testing

R1. To be cost-effective and scalable, test techniques must be automated. This must include both the automation of test case and oracle generation. For regression testing, both test case selection and prioritisation must also be supported.

R2. The user requirements for a test technique must be realistic. It must account for what can be realistically expected given the complexity of systems, the skills and education of software engineers, international standards, and the maturity of supporting technology in the foreseeable future.

R3. Systems must be designed to be testable if any testing technique is to be cost-effective. Methodologies for supporting the design and assessment of testability must be provided. Testability is typically defined as reflecting two dimensions [29]: observability and controllability. One must be able to observe the state of the SUT and set it in a state appropriate for preparing the execution of test cases. This typically entails that built-in test interfaces be provided when designing software components and subsystems [22]. Another important aspect is that the design of a system must enable a rational integration strategy by allowing stepwise component and subsystem integration while minimising the need for stubs or mocks [21].

Analysis of Early Specification and Design Artefacts

R4. Though the analysis of all interesting properties is unlikely to be fully automated in practice, effective decision support should be provided to facilitate the analysis of large specification and design models. The involvement of the analyst must be minimised as well as the amount of information that must be processed to inspect the artefact and achieve a decision.

R5. Analysis techniques, just as for testing, must be realistic in terms of the inputs required from the analyst. This entails that the modelling notation used and underlying technology be carefully selected to be usable for large systems, be supported by effective, extensible, and open architecture tools, and account for international modelling standards. For modelling to be scalable, such notation should effectively support hierarchical, partial, and incremental modelling, the definition of model aspects (cross-cutting concerns at the modelling level), and the automated consistency and completeness checking of different modelling views (e.g., different UML diagrams).

28.4 Moving Forward

This section will outline what is considered to be an ambitious yet realistic research approach to converge towards cost-effective engineering solutions for the verification of software systems within the framework of MDD. The choices made will be explicitly linked to the requirements stated previously.

Model-Based Test Generation and Context Simulation

The point of model-based test generation is to exploit specification or design information for the purpose of automating test case generation (R1) according to systematic strategies and thus check the conformance between an implementation and its specification and design. In order to provide effective and scalable test automation, there is little alternative to model-based testing. Indeed, adequate abstract representations (models) of the SUT must be defined to support the automated derivation of test cases and oracles based on explicitly defined test strategies. Such automated support would be difficult to conceive based on source code analysis[3] or informal, and therefore ambiguous and probably incomplete, textual documentation. With test-ready system models, or in short, test models, test automation is bound to be limited to test execution and replay. On the other hand, as discussed in the section about choice of modelling paradigm on page 428, the test modelling requirements must be realistic so as to be applicable in practice in the context of large and complex software systems. A balance must be struck between the effort invested in modelling and the benefits achieved through verification automation (R2).

Figure 28.3 presents an overview of the activities and artefacts involved in model-based testing. As one can see, the approach involves four parts, respectively related to test modelling from specification artefacts[4], exploiting the test models for test case generation, generating the executable test harness (e.g., scripts) for a targeted platform, and running the test cases on it. Automating test oracles is also a very significant, practical problem throughout the software industry and is discussed in the next subsection, since it implies specific techniques.

[3] White-box testing has been shown not to scale up, as source code static analysis leads to numerous difficulties on non-trivial programs, for example, infeasible execution paths. Furthermore, a great deal of relevant information is difficult to reverse-engineer from source code, for example, states and their invariants.

[4] We can also focus on design artefacts, depending on the purpose and level of testing, though in the context of object-oriented analysis and design (OOAD), design models are refinements of analysis (specification) models. We will only refer to specifications in the remainder of the text.

Let us now examine the main features of Figure 28.3. Test modelling requires devising an analysable model for the specific purpose of test automation from system specifications. If such specifications are informal, then the modelling process is mostly manual, though it can still be supported by a specific modelling methodology and tool. For example, one may derive some form of UML state machines from the informal specification of a control or reactive system. The model then has to be checked for consistency and completeness. Traceability information between the specifications and the test model must be saved so that changes to the specifications can later be more easily accommodated by changes in the test model. The second phase is then to exploit the test model to generate test requirements (e.g., paths to cover all transitions in a state machine [22]), then test cases once the test requirements have been validated (e.g., all transition paths are feasible), and then possibly prioritise test cases if the resulting test suites are large. Prioritisation is usually based on a risk model and can be achieved through some form of optimisation strategy. The risk model can be based, for example, on how safety critical the functions triggered by test cases are (e.g., transition paths) or how error prone the executed components are (e.g., based on historical data or complexity measurements [17]). Last, any model-based testing tool needs to be coupled with the available test script generator and test execution environments on the specific development or deployment platform.

In the context of embedded real-time systems, it is crucial to perform as much verification as possible on the development platform. Running test cases on the deployment platform may be vastly expensive and in many cases dangerous and difficult to set up. To be able to run, say, a control system on a development platform, its environment needs to be simulated. For example, the behaviour sensors and actuators, external systems, or even users need to be emulated as if the system under test were actually running in its deployment environment. One interesting approach to be investigated is the use of UML modelling and extensions to model the system *context* [57] so as to be able to generate the emulation code automatically. One can, based on appropriate models, replace the drivers' code with emulation code having identical interfaces. For example, state machines describing devices can be transformed into code emulating these devices. Using the same modelling technology for modelling the context and the system offers many practical advantages. It must be determined whether, in the context of embedded systems, a UML profile, such as the Profile for Schedulability, Performance, and Time Specification (SPT) [39], could be used to model all relevant properties of a context (e.g., time properties of sensors). The principles of such an approach are illustrated in figure 28.4. Another interesting opportunity to investigate is then to use the context models to drive system testing by adapting a guided random testing approach [3]. Random test generation can be supported by different forms of guidance in order to generate tests automatically in a way that achieves proper coverage of the system and provides sufficient evidence of its reliability.

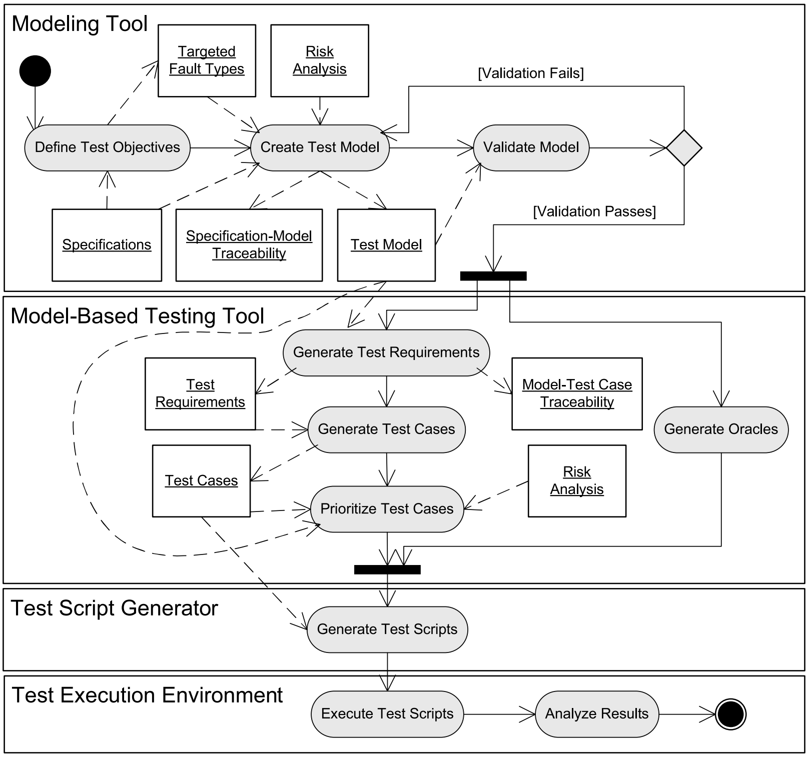


Figure 28.3 A high-level view of model-based testing.

Deriving Test Oracles from Models

One of the hardest problems in test automation is the oracle problem (R1). When a large number of test cases are generated and run, it is absolute necessary to automate the generation of test verdicts. This is the role of test oracles. Because models capture the expected system behaviour, such information can be used to check the conformance of an implementation with its specification. In UML, behaviour can be modeled in different ways and at various levels of details. State machines capture states, their invariants, and their transitions and are particularly suitable for certain types of systems and components, as in the embedded systems domain. Interaction diagrams capture possible component interaction patterns and can be useful during integration testing to check whether observed interactions (e.g., from execution traces) are consistent with expectations. Various types of constraints, defined, for example, with the Object Constraints Language (OCL)—a component of the UML—can be checked at run time, though it is important to consider the execution overhead incurred and its consequences in real-time distributed systems. Examples of types of constraints include operation contracts, class invariants, and safety properties. Research has been carried out to capture execution traces in distributed systems [34]

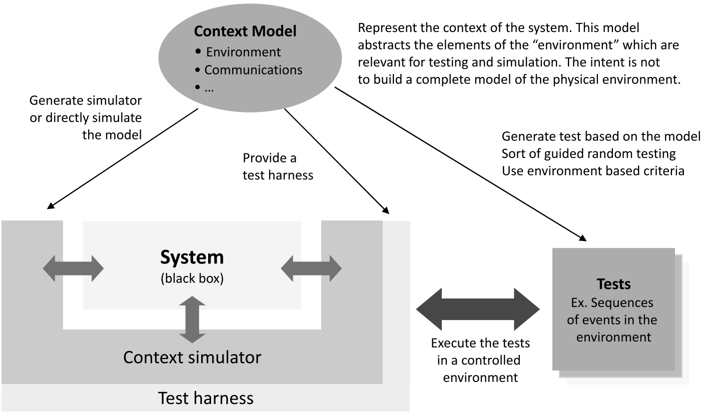

Context Model
• Environment
• Communications
• ...
Represent the context of the system. This model abstracts the elements of the "environment" which are relevant for testing and simulation. The intent is not to build a complete model of the physical environment.
Generate simulator or directly simulate the model
Provide a test harness
Generate test based on the model
Sort of guided random testing
Use environment based criteria
System
(black box)
Context simulator
Test harness
Execute the tests in a controlled environment
Tests
Ex. Sequences of events in the environment

Figure 28.4 Context modelling and simulation for test purposes.

and abstract sequence diagrams from these. One challenge, however, is to obtain models at similar levels of abstraction as design models from execution traces to facilitate comparisons. Studies have assessed the effectiveness of contract assertions as oracles [13] and the necessary level of detail and density of oracles [12] to achieve effective fault detection results. The effectiveness of using state invariants as oracles has also been investigated and has shown in many cases to be insufficient by itself [21, 18]. Overall, very little empirical research and scant results exist regarding the cost-effectiveness of various oracle strategies, whether for model-based testing or in general. Effective model-based oracle strategies and their empirical assessment are therefore an important research endeavour. One particularly difficult area of investigation is the automation of oracles for detecting quality-of-service problems, for example, related to response time, throughput, or security.

Choice of Modelling Paradigm

Ideally, the same modelling paradigm (notation, process) and technology should be used for system modelling and test modelling. In practice, this facilitates integration of specification, design, and testing activities. Designers and testers can then "speak" the same "language" and use common development platforms and technologies. This is of high practical importance, since the lack of collaboration and communication between design and testing teams is a notoriously common problem. As a result, the adopted modelling paradigm needs to account for the needs of all stakeholders, including analysts, designers, and testers.

In the foreseeable future, MDA and UML will remain the de facto international standards for MDD (R2 and R5). They will continue to evolve under the control of the OMG and will be increasingly supported by open-source or open architecture technologies such as Eclipse-based modelling platforms [25]. These tools typically enable the definition of new profiles (e.g., for specific verification purposes) and plug-ins to automate model analysis and testing (R1 and R4). These characteristics are important because they enable the use of common environments for analysts, designers, and testers. Second, by supporting the definition of profiles in a UML context, they enable the extension of design models for the purpose of deriving test models amenable to automation. Through plug-in mechanisms, modelling and development environments can then be extended to automate model-based testing. Many useful UML profiles have already been defined and approved by the OMG, such as profiles for real-time and concurrent systems–Schedulability, Performance, and Time Specification (SPT) [39] and Modeling and Analysis of Real-Time and Embedded Systems (MARTE) [48]–quality of service [38], and testing [15]. All development activities can then be centred around one model repository that is exploited for various purposes, such as the generation of specification and design documents, code generation, test case generation, and model analysis. Furthermore, substantial research is currently underway to adapt the concepts of aspect-oriented development to the UML modelling realm (R5), thus facilitating the definition of cross-cutting properties, for example, safety or security [2]. The current definition of an executable subset of UML [44] is also expected to facilitate model analysis automation.

Testability

In order to decrease the cost of testing and facilitate its automation, providing ways to assess and improve the testability (R3) of architectures and designs is an important endeavour. Work has already been reported on the analysis of dependencies among components in order to identify integration hotspots and devise optimal integration orders so as to minimise testing efforts [21]. Such dependency analyses can be conveniently based on UML models (e.g., class and sequence diagrams, OCL constraints) or extensions providing more detailed information about dependencies. Testability measurement frameworks have also been proposed to help assess testability based on fault injection [41] and object-oriented designs [29]. More research, however, is still required in this important area to guide designers towards testable architectures and designs. In particular, field studies must be carried out to really understand what the most problematic testability factors are. Case studies are also required to understand the cost-benefit relations of various approaches to improve observability, for example, the cost of developing built-in test interfaces and the various trade-offs that can be made in terms of the granularity of information flowing through such interfaces. For example, should concrete object states be made visible or are abstract state assertions sufficient? Also, how does one effectively deal with observability in the context of distributed systems with distributed state information?

Search-Based Model Analysis of Non-Functional Properties

Once models of the systems are available during the specification or design stages, they can be exploited to analyse relevant properties related to non-functional aspects of the system such as safety, security [10], availability, and performance [53]. Such models, in the context of the previous discussion, would typically be expressed with one or several of the existing UML profiles or rely on a newly defined, domain-

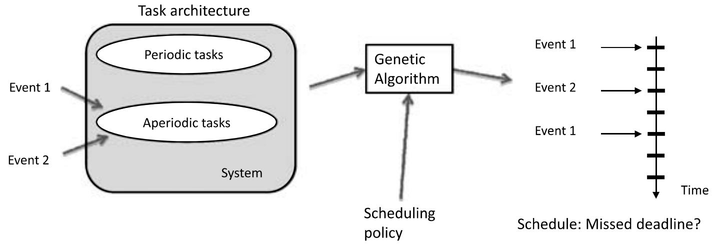


Figure 28.5 Searching for deadline misses.

specific profile. Model analysis is complementary to testing since it can help identify problems early on in the development process, long before any code is available for testing. As described previously, one very active field is that of model checking, which is based on a systematic exploration of the state space of a system as represented by a formal model, for example, expressing real-time properties in temporal logic. The approach adopted here, which will be explored further in the future, is different and likely to be more scalable and applicable, especially in the context of evolving UML and MDA standards. Based on adequate profiles of the UML, a search-based approach to the analysis of non-functional properties of software designs is adopted. What this means is that any non-functional property analysis may be expressed as a search problem and meta-heuristic search algorithms, like evolutionary algorithms, may be used to explore the space of execution scenarios defined by the system model. For example, based on a model of the task architecture of a concurrent real-time system, one can search whether possible scenarios can trigger response time or concurrency problems, such as deadline misses or deadlocks. This is illustrated in figure 28.5, where a task architecture describing tasks, their deadlines, interdependencies, and estimated execution times is provided as input to a genetic algorithm (GA). Assuming a specific scheduling policy, the search is then guided towards schedules (e.g., task seeding times) that minimise the difference between estimated task completion times and their deadlines. The task architecture information can come, for example, from UML design models using the MARTE profile.

Recent work on this topic, using dedicated GAs as a search mechanism, has yielded promising results [52, 7] and has been shown to be very effective, for example, when compared to similar model-checking studies. A search approach does not, of course, guarantee that any property holds in a design. It merely indicates that, if no violations are found, they are unlikely to occur. This is also the case, however, with a model-checking approach in practice, since models can be erroneous anyway and heuristics must be adopted to help improve the scalability of the state exploration [31].

One advantage with a search-based approach is that it does not attempt to perform an exhaustive and systematic exploration of the state space but, rather, searches, in a random but guided way, for specific problems, such as the violation of safety or concurrency properties. In addition, as opposed to model checkers requir-

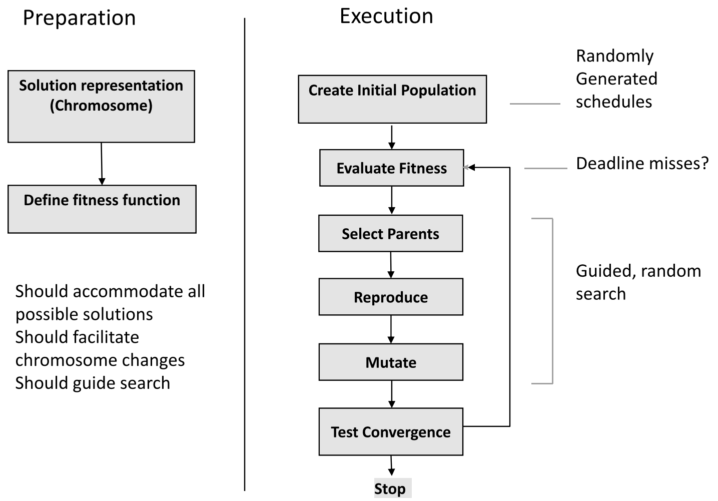

Preparation
Execution
Solution representation (Chromosome)
Define fitness function
Should accommodate all possible solutions
Should facilitate chromosome changes
Should guide search
Create Initial Population
Evaluate Fitness
Select Parents
Reproduce
Mutate
Test Convergence
Stop
Randomly Generated schedules
Deadline misses?
Guided, random search

Figure 28.6 Using GAs to search for deadline misses.

ing the definition of properties in formal logical expression, for example, temporal logic [31], a search-based approach has shown to work in combination with UML models and their extensions [52, 7]. It is important that a model analysis technique avoid or minimise the amount of additional modelling that must be performed, in addition to what is necessary for design and analysis purposes (R5). Ideally, design models, for example, in UML, should be reused or, at worst, augmented to enable analysis. For instance, recent work [7] on detecting deadlocks and starvation problems relied on the SPT and MARTE UML profiles and obtained very encouraging results. This implies, of course, that an appropriate solution representation (defining the search space) and fitness function be defined to effectively guide the search. This may, however, turn out to be impossible for some non-functional properties and clearly identifying the opportunities and limits of such an approach is part of the needed research. As an example, for GAs a number of important decisions have to be made that can potentially impact the efficiency and effectiveness of the search, as illustrated in when searching for deadline misses in real-time systems. In the preparation stage, solutions (e.g., schedules) have to be represented as “chromosomes”, a fitness function must be defined based on an effective search heuristic, and several parameters of the GA must be set appropriately (e.g., mutation and cross-over rates). During the execution stage, the GA then generates a random initial population of chromosomes and then iteratively modifies them through generations of successive populations in what constitutes a random guided search.

Because there is no guarantee that a search mechanism will find property violations, search algorithms need to be carefully assessed through empirical studies, showing that the search is indeed effective at finding problems and scalable to realistic models [40]. Another challenge is to define or adapt appropriate UML profiles or other domain-specific modelling languages to capture all the required information for the search to be effective, but in a way that is consistent with engineering practices, based on international standards and well supported by tools. A technology requiring improbable inputs is not likely to ever transfer to practice.

Empirical Studies of Model-Based Analysis and Testing

Regardless of the specific choice they entail in terms of the test model, coverage criteria, and oracle, all verification techniques that scale up are inherently heuristic: They do not guarantee the detection of faults. Their main point is to be systematic in the way the system or model is exercised and verified so as to obtain a predictable result in terms of fault detection and focus on certain types of faults. Software engineers are therefore confronted with making the difficult decision of choosing a set of test and analysis techniques–possibly different across various test phases–which will fit within their budget and time constraints and that are likely to be effective at detecting faults early in the development process. For example, Briand et al. [45] used simulation to assess and compare various test strategies based on state machines in terms of fault detection. The results showed significant variations across SUTs due to their real-time characteristics, among other things.

It is common to refer to the cost-effectiveness of a technique as the fault detection obtained over the cost of applying it or, even better, the effort saved by fault detection minus the effort of detection. Since one cannot analytically assess or compare the cost-effectiveness of various testing techniques, it is natural to resort to empirical studies. Such studies should investigate the following categories of questions:

- What cost and fault detection rates can be expected from using a verification technique?
- How do alternative techniques compare in terms of both cost and fault detection rates?
- Is it beneficial to combine two or more verification techniques? Are these techniques complementary in terms of fault detection?

Empirical studies are particularly complex, however, because the cost-effectiveness of a verification technique depends on many other factors. For example, regarding testing techniques (see figure 28.7), it is not just the coverage criterion and oracle that determine cost and effectiveness:

- The test technique's degree of automation obviously affects its cost and, under time constraints, its effectiveness, since the level of coverage achieved may be less than 100 per cent. Automation includes the identification of test requirements, the generation of test drivers and test stubs (e.g., mock objects), and the implementation of test oracles.
- The types of faults present in the SUT and their probability of detection (fault profile) affect the test technique's detection capability. For example, in a study investigating the cost-effectiveness of statechart-based testing [18], we found that many faults in a concurrent cruise control could not be easily detected with just test cases generated from statecharts, since the concurrent real-time behaviour of the SUT was not part of the test model and therefore not fully exercised. As

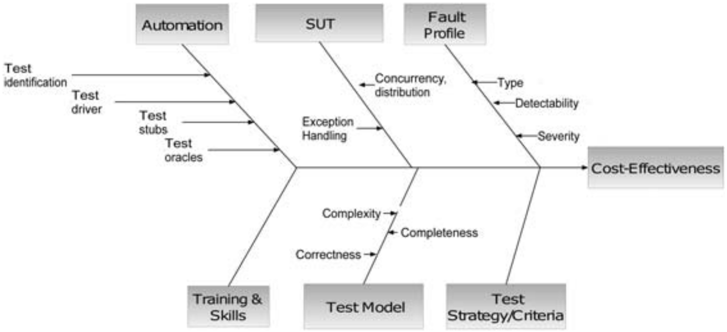


Figure 28.7 Testing cost-effectiveness factors.

a result, certain faults that could only be exercised by a certain scheduling of external events never triggered a failure.

- The SUT itself has, of course, an impact on the probability with which certain faults can be detected given a certain test technique. Concurrency, distribution, and complex exception handling partially determine not only what types of faults are present in the SUT but also how easy these faults will be to detect.
- The training and skills of testers also have a strong impact, since test techniques are usually not entirely automated and require at the very least human input. For example, test models are devised by software engineers and may vary greatly in terms of correctness and completeness. Certain techniques are entirely based on human intuition and an understanding of the system's behaviour, such as the category-partition method [9], and have been observed to yield a wide variety of results under identical conditions [8].
- Test models vary in complexity and cost. Certain techniques require complex models that entail significant cost. Furthermore, such models are likely to be incorrect or even incomplete in practice, which is an important aspect regarding how human factors can affect a technique's applicability.

Because the preceding factors can have a dramatic impact on a test technique's cost-effectiveness, empirical studies should try to control for such factors in order to ensure that any cost-effectiveness comparison among test techniques is unbiased but also representative of the targeted context of the study.

28.5 Software Verification Research at Simula

This section provides a structured overview of research objectives, recent work, and industry collaborations involving the author and his colleagues at Simula.

Research Scope

One Simula Research Laboratory's mandate is to perform industry-driven research to increase SRL's relevance and impact on actual engineering practices. In this context, and based on the discussions in previous sections, our approach to verification research can be characterised as follows:

- We assume that development and test models are expressed in the UML or extensions through profiles. Depending on the targeted aspect of verification, we may be led to define new profiles. We therefore place our work within the MDA standard proposed by the OMG in order to benefit from a rapidly growing, and often open-source, technological base and comply with the only de facto international standard regarding software modelling.
- We model not only the software being tested but also its environment. This is particularly important for embedded systems that need to be verified and tested on the development platform before undergoing the same in realistic settings, since such environment (context) models can help us generate the emulation code we need to emulate drivers and other context elements.
- Model analysis and test generation are automated through the use of metaheuristic search algorithms such as GAs. Our choice is therefore to rely on heuristic search rather than systematic and complete state exploration, such as in model checking. The main motivation is to achieve scalable solutions.
- Simulations, field studies, or controlled experiments are used to assess empirically the cost and fault detection effectiveness of verification strategies, as well as their scalability to larger models. Empirical research is therefore a major component of our activities and requires developing appropriate methodologies to empirically assess verification techniques.
- Particular emphasis will be placed on verifying non-functional properties, including safety, robustness, response time, security, and concurrency. Very little is available regarding these aspects in the context of MDD.

Related Research

To provide the background information that may help explain and exemplify the perspectives described above, recent work by the author related to this chapter can be summarised as follows.

Model Analysis

- Automated traceability of UML model refinements [27]: Traceability between analysis (domain) models and design models must be preserved so as to be able to propagate changes from one to the other. This work provides a means to automate the creation of traceability information in the context of UML model refinements.
- Impact analysis based on UML models [1]: This works provides a solution to identify the ripple effects of changes to the design across the system, based on the analysis of UML models, with a focus on class and sequence diagrams with OCL constraints.
- Model-based prediction of resource usage and load in distributed systems [4]: UML sequence diagrams, augmented with timing information, are used to pre-

dict system behaviour at the design phase in terms of network traffic, CPU, and memory usage.

- Deadlock and starvation analysis based on UML models with the MARTE profile [7]: Based on UML MARTE models (e.g., sequence diagrams), GAs are used to search for deadlocks and starvation problems in concurrent systems.
- Definition of a safety profile to support third-party safety accreditation in the aerospace domain [10]: This work is a first step towards the definition of a safety UML profile to enable safety analysis and accreditation in the context of MDA.
- Reverse-engineering UML sequence diagrams from dynamic analysis in the context of distributed Java systems [34]: This was a first step towards collecting and analysing traces in the context of distributed systems to be able to compare executions to UML design models and identify discrepancies in an automated fashion.

Model-Based Testing

- Regression test selection based on UML models [26]: Regression test selection based on source code analysis is a well-researched area. The only problem is that it is applicable only after changes have been applied to the source code. This work provides a way to assess regression test efforts and identify test cases to rerun based on changes in UML design models.
- Empirical evaluation and improvement of strategies for state machine-based testing and its combination with white-box testing [45, 18]: Simulations and series of experiments were performed to assess the cost-effectiveness of testing strategies based on state machines, how they compare to simple control flow testing, and whether the two should somehow be combined. A careful analysis also led to refinements of testing and oracle strategies in addition to practical recommendations.
- Improving state machine testing with data flow analysis [56]: Because empirical results have shown that it is not easy to select paths to test in state machines, this work investigates whether data flow information, derived from operation contracts and guard conditions in OCL, can be used to select high-fault-detection paths. The authors empirically investigate whether paths that exercise the most data flow also help detect more faults.
- Stress-testing distributed systems [54, 30]: Based on UML sequence diagrams augmented with timing information and specially designed GAs, the goal here is to stress-test distributed systems with test scenarios maximising network traffic.
- Contracts as test oracles in sequential and concurrent contexts [13, 20]: This work extends the Java Modeling Language to allow the definition of contract assertions in the context of concurrent systems. This is a necessary technology to help define test oracles for concurrent Java systems.
- Stress-testing real-time systems by generating test cases that maximise completion times for target tasks [52]: The goal was to stress-test real-time systems to maximise their chances of missing deadlines. The goal was to increase confidence so that, if such test cases were successful, deadline misses would be unlikely. Test cases are derived from UML models augmented using the SPT profile, which

has now been replaced by MARTE. Similar to concurrency problems, deadline misses, in some cases, can be directly identified from the models, without resorting to testing.

- State machine-based integration testing [6]: Based on the analysis of state machines and interaction diagrams in UML models, this work attempts to devise systematic class integration testing strategies.
- Contract-based test automation of commercial off-the-shelf (COTS) components [62]: Assuming only contracts are available to describe the functionally of a COTS component, since design details are usually proprietary, this work adapts existing strategies to automatically derive adequate test suites that can be used by the component users to assess its reliability.

Industry-Driven Engineering Research in Software System Development

It is important to describe how we intend to proceed with the previously mentioned research given SRL's mandate. As opposed to many other engineering disciplines, it is rarely possible for software engineering researchers to reproduce the phenomena they are studying in the laboratory. Despite many decades of research in software engineering, the impact of academic software engineering research on engineering practice is unclear, although there is probably wide variation across different research areas. The gap between research and practice is, in our opinion, partly due to the historical specificities of software engineering, which originally branched out of computer science, which itself was initially a branch of discrete mathematics. There is therefore little engineering tradition in software engineering research and, as a result, research is far too rarely problem driven or based on precisely defined problems reflecting the reality of software system development. Though research on fundamental problems is obviously crucial, such an imbalance contrasts sharply with research in other engineering disciplines. Furthermore, because of the human and organisational factors involved in software development, software engineering cannot simply be seen as a mathematical problem. The right trade-offs have to be found between the seemingly conflicting objectives of engineering rigor, changing requirements, and tight schedules. Any solution must be scalable to large systems and teams, be compatible with the practice of incremental development, and support frequent change.

The preceding statements entail that researchers cannot fully understand the actual problems or devise and assess suitable engineering solutions, whether for verification or any other aspect of software development, if they do not work in close collaboration with industry. This is why software engineering research at SRL has focussed on industry-driven projects. A difficult question, however, is how to make such collaboration effective and productive. The practicing engineer's priority is to complete projects on schedule and not, unless there exists a clear mandate to do so, to improve engineering practices. Since software development is a collective endeavour, the task of implementing change in practices, with its training and mentoring requirements, is also far from a trivial matter and requires dedicated resources.

To make collaborative, industry-driven research effective, the verification group at SRL has adopted a research process that relies on integrating the work of doctoral candidates into the practice of industrial partners. At a high level, the first step of the process is to identify difficult, long-standing problems that our industry partners have been facing on projects. The state of the art related to the identified problem is then assessed by performing a systematic and thorough review of the scientific literature. Usually, significant gaps are found, as many of the proposed techniques are incomplete or simply not applicable as defined, for example, not scalable. A solution is then devised in context and scientifically evaluated on actual systems, using, for example, actual fault data, accounting for human and organisational factors. The last step is to generalise the solution to make it applicable to a wider context by attempting, for example, to relax some of the assumptions underlying the work.

Adopting such an approach to research presents a number of practical challenges that should be addressed. It is important to ensure that the problem selected is significant enough to constitute a doctoral thesis topic. In the current stage of maturity of software engineering practice and research, this is usually not a difficult problem. Second, long-term commitment from the industry partners is required, because such collaborations must necessarily last the duration of a thesis and cannot be interrupted without serious consequences. Working with industry partners requires understanding their problems, technology, and working processes, which entails an effort overhead not normally present in traditional doctoral work.

The following provides two illustrative examples of recent industry collaborations in which the author and his SRL colleagues were involved.

Telecom. This project took place in collaboration with an industry partner in the telecom domain. One problem identified on this project was that insufficient resources were available to thoroughly test all components on every release of a telecom product. As a result, testing was ad hoc and mostly driven by individual choices. We analysed change and fault data, as well as the source code of a number of releases and devised a prediction model to identify, on each release, where faults were more likely to be located. A testing strategy was then devised to adjust the testing intensity to the likelihood of faults in a component. This strategy was assessed and showed a 100 per cent return on investment.

Manufacturing. This project took place in the context of the development of manufacturing systems. One important problem raised was related to the safety of such systems and therefore the verification of a safety component in charge of monitoring unsafe events during the system's execution. Given the stringent safety requirements, a model-based testing approach, based on formal state models, was defined. To support change, a design strategy was elaborated to ensure traceability between the state model and the source code, thus facilitating future changes. Empirical studies are underway to assess the benefits of both the testing and design approaches.

28.6 Conclusions

This chapter addresses the verification of software systems, a highly crucial topic, given the growing importance of software throughout most economic sectors, especially in the many domains where it plays a safety- or business-critical role. This chapter argues that the current state of the art is not even close to addressing the highly challenging problems that software engineers are facing when verifying software systems. The scale and complexity of the work are so daunting that automation is a requirement. Effective automation, however, can only be achieved if systems are described by appropriate models that not only help describe their specification and design but also support verification, including both analysis of specification and design artefacts and model-based testing. With the advent of the international UML 2.0 standard and its associated MDA framework, a growing body of technology, including open-source and architecture tool platforms, has rapidly developed in recent years. This means that the introduction of MDD practices, including model-driven verification, is a much more realistic opportunity today than even five years ago.

Through carefully adapted and tailored modelling technologies, with a rigorous scientific approach, and through tight collaborations between industry and research institutions, the complexity and scale of software verification may now be tackled. This chapter highlights the need for a stronger focus on non-functional aspects of verification and addresses the use of meta-heuristic search algorithms as a practical alternative to automated model analysis and model-based testing. It also shows that model-based verification is essentially a trade-off between modelling effort in the early stages of development and automation gains in verification activities. Though devising scalable and effective solutions for model-driven verification is a significant challenge, with the rising complexity and criticality of software-based systems the use of models is bound to yield increased benefits. The benefits, costs, and scalability of the proposed solutions can only be investigated in industrial contexts and on actual systems, accounting for human, organisational, and economic factors. Such investigations are best done through carefully designed empirical studies involving both SRL researchers and their industrial partners. Such a research agenda is therefore in perfect alignment with the mandate of SRL, a leader in industry-driven, high-impact IT research.

References

[1] L. C. Briand, Y. Labiche, L. O'Sullivan, and M. Sowka. Automated impact analysis of UML models.*Journal of Systems and Software*, 79(3):339–352, 2006.

[2] R. France, I. Ray, G. Georg, and S. Ghosh. Aspect-oriented approach to early design modelling. *IEE Proceedings-Software*, 151(4):173–185, 2004.

[3] P. Godefroid, N. Klarlund, and K. Sen. DART: directed automated random testing. *PLDI '05: Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation*, pages 213–223, New York, NY, USA, 2005. ACM.

[4] V. Garousi, L. Briand, and Y. Labiche. A UML-based quantitative framework for early prediction of resource usage and load in distributed real-time systems. *Software and Systems Modeling (Springer)*, 2008.

[5] IBM. Rational test RealTime. 2005.

[6] S. Ali, L. C. Briand, M. J. Rehman, H. Asghar, M. Z. Z. Iqbal, and A. Nadeem. A state-based approach to integration testing based on UML models. *Information and Software Technology*, 49(11-12):1087–1106, 2007.

[7] M. Shousha, L. Briand, and Y. Labiche. A uml/spt model analysis methodology for concurrent systems based on genetic algorithms. *ACM/IEEE 11th International Conference in Model Driven Engineering Languages and Systems (MODELS 2008)*, 2008.

[8] T. Y. Chen, P. L. Poon, S. F. Tang, and T. H. Tse. On the identification of categories and choices for specification-based test case generation. *Information and software technology*, 46(13):887–898, 2004.

[9] T. J. Ostrand and M. J. Balcer. The category-partition method for specifying and generating fuctional tests. 1988.

[10] G. Zoughbi, L. C. Briand, and Y. Labiche. A uml profile for developing airworthiness-compliant (rtca do-178b) safety-critical software. *International Conference on Model Driven Engineering Languages*, 2007.

[11] C. Ghezzi, M. Jazayeri, and D. Mandrioli. Fundamentals of software engineering. 1991.

[12] Y. L. Traon, B. Baudry, and J. M. Jezequel. Design by contract to improve software vigilance. *IEEE Transactions on Software Engineering*, 32(8):571, 2006.

[13] L. C. Briand, Y. Labiche, and H. Sun. Investigating the use of analysis contracts to improve the testability of object oriented code. *Software Practice and Experience*, 33(7), 2003.

[14] G. Tassey. The economic impacts of inadequate infrastructure for software testing. *National Institute of Standards and Technology RTI Project*, 2002.

[15] P. Baker, Z. R. Dai, J. Grabowski, Ã. Haugen, S. Lucio, E. Samuelsson, I. Schieferdecker, and C. Williams. The UML 2.0 testing profile. *8th Conference on quality Engineering in Software Technology*, Nuremberg, Germany, 2004.

[16] B. F. Jones. Special issue on metaheuristic algorithms in software engineering. *Information and Software Technology*, 43:14, 2001.

[17] N. Fenton and S. L. Pfleeger. *Software metrics: a rigorous and practical approach.* PWS Publishing Co. Boston, MA, USA, 2nd edition, 1998.

[18] S. Mouchawrab, L. C. Briand, and Y. Labiche. Assessing, comparing, and combining statechart-based testing and structural testing: An experiment. *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2007.

[19] E. Weyuker, T. Goradia, and A. Singh. Automatically generating test data from a boolean specification. *IEEE Transactions on Software Engineering*, 20(5):353–363, 1994.

[20] W. Araujo, L. Briand, and Y. Labiche. Concurrent contracts for java in JML. *Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on*, pages 37–46, 2008.

[21] L. C. Briand, Y. Labiche, and Y. Wang. An investigation of graph-based class integration test order strategies. *IEEE Transactions on Software Engineering,* 29(7):594–607, 2003.

[22] R. Binder. *Testing object-oriented systems: models, patterns, and tools.* Addison-Wesley Professional, 1999.

[23] R. V. Binder. Testing object-oriented software: a survey. *Software Testing, Verification and Reliability,* 6(3–4):125–252, 1996.

[24] T. Pender *UML bible.* John Wiley & Sons, Inc. New York, NY, USA, 2003.

[25] E. Foundation. Eclipse modeling framework (EMF). May 2005.

[26] L. Briand, Y. Labiche, and S. He. Automating regression test selection based on uml designs. *Information and Software Technology (Elsevier),* 51(1), 2009.

[27] L. C. Briand, Y. Labiche, and T. Yue. Automated traceability analysis for UML model refinements. *Information and Software Technology,* 51(2):512–527, 2009.

[28] M. J. Harrold, J. A. Jones, T. Li, D. Liang, A. Orso, M. Pennings, S. Sinha, S. A. Spoon, and A. Gujarathi. Regression test selection for java software. *Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications,* pages 312–326. ACM New York, NY, USA, 2001.

[29] S. Mouchawrab, L. C. Briand, and Y. Labiche. A measurement framework for object-oriented software testability. *Journal of Information & Software Technology,* 47(15):979–997, 2005.

[30] V. Garousi, L. C. Briand, and Y. Labiche. Traffic-aware stress testing of distributed systems based on UML models. *Proceedings of the 28th international conference on Software engineering,* pages 391–400. ACM New York, NY, USA, 2006.

[31] M. Pezzé. *Software Testing and Analysis: Process, Principles and Techniques.* Wiley, 2008.

[32] A. C. D. Neto, R. Subramanyan, M. Vieira, and G. H. Travassos. A survey on model-based testing approaches: a systematic review. *Proceedings of the 1st ACM international workshop on Empirical assessment of software engineering languages and technologies: held in conjunction with the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE) 2007,* pages 31–36. ACM New York, NY, USA, 2007.

[33] J. H. Andrews, L. C. Briand, Y. Labiche, and A. S. Namin. Using mutation analysis for assessing and comparing testing coverage criteria. *IEEE Transactions on Software Engineering,* 32(8):608, 2006.

[34] L. C. Briand, Y. Labiche, and J. Leduc. Towards the reverse engineering of uml sequence diagrams for distributed java software. *IEEE Transactions on Software Engineering,* 32(9), 2006.

[35] W. E. Wong, J. R. Horgan, A. P. Mathur, and A. Pasquini. Test set size minimization and fault detection effectiveness: A case study in a space application. *The Journal of Systems & Software,* 48(2):79–89, 1999.

[36] G. Rothermel and M. J. Harrold. A safe, efficient regression test selection technique. *ACM Transactions on Software Engineering and Methodology (TOSEM),* 6(2):173–210, 1997.

[37] N. Juristo, A. M. Moreno, and S. Vegas. A survey on testing technique empirical studies: how limited is our knowledge. *Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n*, pages 161–172, 2002.

[38] U. OMG. Profile for modeling quality of service and fault tolerance characteristics and mechanisms. *Object Management Group*, 2005.

[39] U. OMG. Profile for schedulability, perfomance and time specification. 2005.

[40] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege. A systematic review of the application and empirical investigation of evolutionary testing. Technical report, Simula Research Laboratory, 2008.

[41] J. Voas, M. Schmid, M. Schatz, and D. Wallace. Testability-Based assertion placement tool for Object-Oriented software. *NASA*, (19980045759), 1998.

[42] B. Marick. *The craft of software testing*. PTR Prentice Hall Englewood Cliffs, NJ, 1995.

[43] G. Rothermel, R. H. Untch, C. Chu, and M. J. Harrold. Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, 27(10):929–948, 2001.

[44] S. J. Mellor and M. Balcer. *Executable UML: A foundation for model-driven architectures*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2002.

[45] L. C. Briand, Y. Labiche, and Y. Wang. Using simulation to empirically investigate test coverage criteria based on statechart. *Proceedings of the 26th International Conference on Software Engineering*, pages 86–95. IEEE Computer Society Washington, DC, USA, 2004.

[46] R. Conradi, P. Mohagheghi, T. Arif, L. C. Hegde, G. A. Bunde, A. Pedersen, and E. Norway-Grimstad. Object-Oriented reading techniques for inspection of UML models– an industrial experiment.

[47] L. C. Briand, M. D. Penta, and Y. Labiche. Assessing and improving state-based class testing: a series of experiments. *IEEE Transactions on Software Engineering*, 30(11):770–783, 2004.

[48] OMG. The official OMG MARTE website. 2008.

[49] D. Lee and M. Yannakakis. Principles and methods of testing finite state machines-A survey. *Proceedings of the IEEE*, 84(8):1090–1123, 1996.

[50] O. M. G. Uml. 2.0 superstructure specification. *OMG ed*, 2003.

[51] A. G. Kleppe, J. Warmer, and W. Bast. *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2003.

[52] L. C. Briand, Y. Labiche, and M. Shousha. Using genetic algorithms for early schedulability analysis and stress testing in real-time systems.*Journal of Genetic Programming and Evolvable Machines*, 7(2), 2006.

[53] D. C. Petriu. Performance analysis with the SPT profile. pages 205–224, 2005.

[54] V. Garousi, L. C. Briand, and Y. Labiche. Traffic-aware stress testing of distributed real-time systems based on uml models using genetic algorithms.*Journal of Systems and Software*, 81(2):161–185, 2008.

[55] J. A. Jones and M. J. Harrold. Empirical evaluation of the tarantula automatic fault-localization technique. *Proceedings of the 20th IEEE/ACM international Con-*

ference on Automated software engineering, pages 273–282. ACM New York, NY, USA, 2005.

[56] L. C. Briand, Y. Labiche, and Q. Lin. Improving statechart testing criteria using data flow information. *16th IEEE International Symposium on Software Reliability Engineering, 2005. ISSRE 2005*, page 10, 2005.

[57] H. Gomaa. *Designing Concurrent, Distributed, and Real-Time Applications With Uml.* Addison-Wesley Professional, 2000.

[58] B. F. Jones, H. H. Sthamer, and D. E. Eyres. Automatic structural testing using genetic algorithms. *Software Engineering Journal*, 11(5):299–306, 1996.

[59] T. S. Chow. Testing software design modeled by finite-state machines. *IEEE Transactions on Software Engineering*, pages 178–187, 1978.

[60] C. Yilmaz, M. B. Cohen, and A. A. Porter. Covering arrays for efficient fault characterization in complex configuration spaces. *IEEE Transactions on Software Engineering*, 32(1):20–34, 2006.

[61] P. Ammann and J. Offutt. *Introduction to software testing.* Cambridge University Press, 2008.

[62] L. C. L. Briand, Y. Labiche, and M. Sowka. Automated, contract-based user testing of commercial-off-the-shelf components. *ACM/IEEE International Conference on Software Engineering (ICSE).* ACM Press, 2006.

29

THE INDUSTRY IS OUR LAB — ORGANISATION AND CONDUCT OF EMPIRICAL STUDIES IN SOFTWARE ENGINEERING AT SIMULA

Dag I. K. Sjøberg

Dag I. K. Sjøberg
Simula Research Laboratory

Dag I. K. Sjøberg
Department of Informatics, University of Oslo, Norway.

Sjøberg served as head of the Software Engineering department from the start in 2001 till June 2008. He has now left, and is currently affiliated to the University of Oslo.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_29,

PROJECT OVERVIEW

Software Engineering Research Methods

Modern-day society is completely dependent on reliable and efficient software. Billions of dollars are squandered yearly on poorly planned and inefficiently executed software development projects. As stated in the strategy of Simula's Software Engineering department, the motivation for the research conducted by this department is to support the private and public software industry in developing higher-quality systems with improved timeliness in a more cost-effective and predictable way.

Scientific challenges

Software development requires the use of appropriate technology (processes, methods, and tools). The overall goal of the SE department is to evaluate the appropriateness of such technology in various industrial settings. At present, most software engineering technology is the subject of only anecdotal evidence or claims or argumentation based on no empirical evidence. And when empirical studies are conducted, they mostly involve students as subjects in artificial settings. Thus, the major challenge the department (as well as the empirical software engineering research community as a whole) faces is to conduct studies of software engineering technologies that are scientifically sound and at the same time convincing to practitioners and managers in industry.

Obtained and expected results

The SE department's strategy of investing in experiment infrastructure and hiring professionals as subjects has contributed to a significant improvement in the state of the art regarding the realism and scale of experiments and control in real-life case studies. During its first eight years, Simula conducted studies whose participants included about 6000 professional software developers and managers from 273 companies in 24 countries. There are indications that this increase in realism, scale, and control has made the results more reliable and relevant to industry than is the case for most studies in software engineering. Increasing the realism of tasks and systems in experiments and control in case studies is, however, still a challenge for Simula and the rest of the research community.

THE INDUSTRY IS OUR LAB — ORGANISATION AND CONDUCT OF EMPIRICAL STUDIES IN SOFTWARE ENGINEERING AT SIMULA

29.1 Introduction

Software systems form the foundation of modern information society and many of those systems are among the most complex things ever created by man. The quotation that follows is from the 1999 President's Information Technology Advisory Committee Report[1], but is as valid today as it was then:

> Our ability to construct... needed software systems and our ability to analyze and predict the performance of the enormously complex software systems that lie at the core of our economy are painfully inadequate. We are neither training enough professionals to supply the needed software, nor adequately improving the efficiency and quality of our construction methods.

Software engineering is about developing, maintaining, and managing high-quality software systems in a cost-effective and predictable way. Software engineering research studies the real-world phenomena of software engineering and concerns the development of new technologies (process models, methods, techniques, tools, or languages), or the modification of existing ones, to support software engineering activities; and the evaluation and comparison of the effect of using such technology in the often very complex interaction of individuals, teams, projects, and organisations and various types of tasks and software systems. Sciences that study real-world phenomena, that is, empirical sciences, necessarily use empirical methods as their principal mode of inquiry. Hence, if software engineering research is to be scientific, it too must use empirical methods (using logic to reason about the behaviour of software engineers is generally infeasible). Empirical research seeks to explore, describe, predict, and explain natural, social, or cognitive phenomena by obtaining and interpreting evidence through, for example, experimentation, systematic observation, interviews or surveys, or by the careful examination of documents or artefacts.

The motivation to focus on industry-relevant research is reflected in the name of the research group we formed at the University of Oslo in 1999, Industrial Systems Development, and further in the motto of the SE department at Simula Research

[1] www.ccic.gov/ac/report

Laboratory, formed in 2001, “The industry is our lab”, which reflects that research in the field of informatics or computing is meaningless unless we keep its applications squarely in sight.

A particular challenge when conducting empirical studies in software engineering is that we want to produce results that are scientifically sound yet at the same time relevant to engineers and managers in the software industry. Hence, there is a trade-off between *control*, for identifying cause-effect relations, and *realism*, for making the results transferrable to industrial applications. This chapter will report on Simula’s effort to improve the state of the art along these two dimensions.

The remainder of the chapter is organised as follows. Section 29.2 gives an overview of the various ways in which that the SE department at Simula has interacted with industry. Section 29.3 describes the incentives we gave to industry to collaborate with us. Section 29.4 describes the sophisticated experiment infrastructure without which it would hardly have been possible to conduct many of our comprehensive experiments. We then focus on how the department has contributed to improving the state of the art regarding realism in experiments (section 29.5) and control in case studies (section 29.6). Section 29.7 presents our conclusions.

29.2 Forms of Collaboration With Industry

Collaboration between a research institution and a company in industry can take many forms. Note here that we interpret the term *collaboration* in a wide sense, to include many kinds of more or less formal interactions. Table 29.1 shows the numbers of companies and persons that have been involved in various types of collaboration with Simula from 2001 to 2008. (Note that the term company also includes public service units.) The companies column refers to unique companies within each category. The people involved may not be unique, however, because we did not record their identities as individuals. Table 29.1 shows that the majority of the companies have taken part in experiments organised by Simula.

Included under the category of joint research in table 29.1 are the processes of writing grant applications and working actively together in research projects that have received grants. Writing scientific articles together with people from industry reflects close collaboration. Eight companies have had employees co-author Simula articles, but about half of the articles were written in collaboration with employees of a single company. This was made possible by the fact that that company has its own research department and has had a long-lasting relationship with Simula.

The transfer of knowledge and technology takes many forms at Simula. It ranges from in-depth collaborations on joint research projects to teaching at a university (where quite a few of the students also work in industry) to writing feature articles in IT magazines. The principal means of transferring knowledge and technology in the SE department to industry, however, has been through courses and seminars tailored and given to industry. To make the contents of the courses that the department teaches at the University of Oslo relevant to industry, we have invited software engineers and managers to give guest lectures at both the undergraduate and postgraduate levels. Finally, to support the scale, efficiency, and quality of many of our studies, we hired external consultants to build experiment infrastructures.

Type of collaboration		Companies	Persons
Empirical studies with professional practitioners as participants	Experiments (from 10 minutes to 2 weeks)	262	4330
	Case studies (a few days to 3 years)	8	25
	Action research (up to 5 years)	9	84
	Interviews (typically 1 hour)	18	63
Joint research	Grant applications	18	36
	Grants received	29	82
	Co-authoring scientific papers	8	24
Knowledge and technology transfer	Giving courses	4	101
	Giving seminars	33	1189
Teaching SE courses	Guest lectures from industry at courses given by the SE department	10	31
Acquiring consultancy work	For example, to build infrastructure and organise studies	12	22
Total		411 (322 unique)	5987

Table 29.1 Forms of collaboration with industry.

29.3 Incentives

Section 29.2 described various forms of collaboration between Simula and industry. This section focuses on the incentives for the collaborations. First of all, the SE department has conducted controlled experiments on a scale and with a realism never before seen in the context of software engineering research. Studies of fully realistic development projects, in which several companies develop the same system independently of each other, have also never been seen before in the research community. Crucial to the success of these studies has been the strategy of hiring consultants as participants. We pay the consultancy companies their standard fees for individuals or a fixed price for a complete project, just like any other client. The companies have routines for defining (small) projects with local project management, resource allocation, budgeting, invoicing, the provision of satisfactory equipment, and so forth. It is difficult to find subjects who are employed in an in-house software development company, because management typically will focus all effort on the next release of their product.

Table 29.2 shows that by December 2008, Simula paid 158 companies to provide a total of 1094 persons to take part in experiments. Nevertheless, most participants in our experiments took part as attendees of Simula's seminars. In these experiments, which are conducted in the field of software effort estimation, participants spend 10 to 30 minutes completing a questionnaire [6]. We also paid consultancy companies for full development projects on which we performed case studies, as well as for extra data collection and interviews in other case studies.

A third model is when industry pays for its own time and the Norwegian Research Council funds the researchers. We successfully applied this model in several software process improvement projects in which action research was applied. Action research attempts to provide practical value to the client organisation while simultaneously contributing to the acquisition of new theoretical knowledge. It can be characterised as "an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning" [16].

Type of collaboration	Incentive	Companies	Persons
Experiments	Simula pays industry	158	1094
	Simula gives seminar/increased knowledge	125	3230
Case studies	Simula pays industry	8	25
Action research	Simula offers expertise, the Norwegian Research Council pays for Simula's time (40%), industry spends its own time (60% of total costs) to improve software/business processes	8	83

Table 29.2 Incentives for collaboration.

In a systematic review of 113 experiments published in the leading journals and conferences in the field over a decade, none of the articles reported that professionals were paid to take part in the experiments. (In three cases, students were paid to take part.) Simula is therefore unique in this respect. One of the reasons we have been able to pay directly to conduct empirical studies, which have cost up to 200,000 Euros for a single study, is that the SE department has fully exploited the unique opportunity to use its resources in an optimal manner. (The fact that the same person manages the research, budget, and administration has made it easier to focus resources to achieve scientific goals than if different people had different responsibilities.) About 20 per cent of the budget has been spent on such studies. This was achieved mainly at the expense of employing a larger number of researchers. In recent years, however, this model has been difficult to sustain due to the joint constraints of inflation and an increase in the number of researchers. To maintain progress at our expected rate, we need to include funding for empirical studies in our research grant applications. One budgets for money for positions, equipment, and travel; there is therefore no reason why money for conducting empirical studies could not be included in research grants. Of course, this means that the grants available in our field must be increased but, given the importance of software systems in society, there is no reason that research projects in SE should be less comprehensive or funded less adequately than large projects in other disciplines, such as physics and medicine.

29.4 Experiment Infrastructures

The logistics involved in running large-scale experiments and other studies with industry are tremendous. The participants must be registered, the experimental materials (e.g., questionnaires, task descriptions, code, and tools) must be distributed to each participant, the progress of the experiment needs to be controlled and monitored, the results of the experiment need to be collected and analysed, the task duration must be recorded, payment information must be collected for hired subjects, and so on. To support these logistics, as well as the automatic recovery of experiment sessions and backup of experimental data, Simula has developed an Internet-based tool, the Simula Experiment Support Environment (SESE) [15], which has been crucial to the success of many of the SE department's experiments (see figure 29.1). It is built on top of a commercial human resource management system. The commercial vendor of this system was also hired to develop SESE.

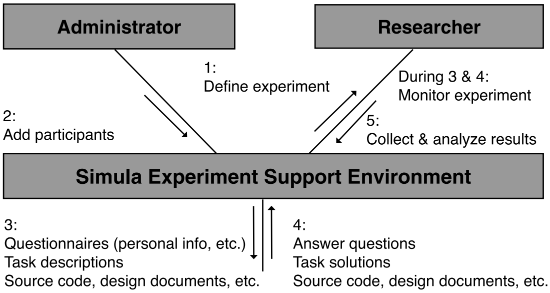

Administrator
Researcher
1:
Define experiment
During 3 & 4:
Monitor experiment
2:
Add participants
5:
Collect & analyze results
Simula Experiment Support Environment
3:
Questionnaires (personal info, etc.)
Task descriptions
Source code, design documents, etc.
4:
Answer questions
Task solutions
Source code, design documents, etc.

Figure 29.1 Internet-based infrastructure to support experiments.

Simula runs studies in many countries (see table 29.2) in order to (1) be able to provide enough subjects with sufficient qualifications for a given study, (2) reduce the costs of hiring professionals, (3) provide a more representative selection of subjects than those limited to a given country, and (4) explicitly study cultural differences in, for example, studies on outsourcing and off-shoring. SESE has been invaluable, both for running distributed experiments in many countries (enabling flexibility regarding location and thus making it easier for professionals to take part in the experiments) and for managing many subjects simultaneously (up to 100 subjects have taken part in an experiment at the same time in one location).

SESE has been extended with a module that supports the collection of qualitative data obtained from software engineering experiments, in particular feedback from subjects during experiments [10]. Such feedback includes useful complementary data to validate data obtained from other sources about the quality and duration of tasks, process conformance, problem-solving processes, problems with experiments, and the subjects' perception of the experiment.

Country	Companies	Persons
Norway	173	3593
India	18	110
Russia	17	36
Sweden	12	146
Ukraine	9	94
Pakistan	7	20
Romania	5	57
Belarus	4	26
Bulgaria	4	8
Nepal	4	101
UK	4	58
Vietnam	3	77
Denmark	2	61
Germany	2	80
Moldova	2	4
Poland	2	7
PR China	1	2
Czech Rep.	1	3
Italy	1	6
Lithuania	1	5
Philippines	1	2
Serbia	1	2
Slovakia	1	2
Thailand	1	2
Total	276	4502

Table 29.3 Companies and people that have participated in studies at Simula.

We realised early in our use of SESE to conduct experiments that the tool could be modified to assess programmer skill in general, which would lead to cost savings and improve the decision-making process in industry. Hence, as early as 2002 we held interviews in seven organisations about their interest in an instrument that would support the assessment of programmers based on actual programming tasks, as opposed to simply answering questions textually or using multiple-choice forms. As a result of the great interest in this issue, we announced a PhD scholarship that should focus partly on basic research in this area and partly on the development of a commercial tool for assessing programming skill. A PhD student is now in the final phase of developing an instrument that includes both a model (a Rasch model, which is frequently used in cognitive psychology) and a prototype tool for such an assessment. We believe that the commercial potential is great. This project has already received more than 100.000 Euros from a Norwegian innovation programme.

SESE has attracted a great deal of interest and is now also being used by other research organisations. One can also envisage many extensions of such an environment. In particular, it would be useful to have an infrastructure that supports experiments lasting longer than one day, many research methods (case studies and action research in addition to the present support for experiments and surveys), usability studies, and integration with other study equipment, including audio and video facilities. The research institute SINTEF has already developed an experiment environment that focus on usability studies. Both Simula and SINTEF are now in the process of writing a research grant application to the Norwegian Research Council with the aim of developing and generalising our existing support environments into a full-scale Internet experimentation facility that would be flexible with respect to technology components, methods and processes, and participants.

29.5 Increased Realism in Experiments

Empirical studies usually involve a trade-off between control and realism [5]. The classical method for identifying cause-effect relations is to conduct controlled experiments (called simply experiments below) where only a few variables vary. A common criticism of software engineering experiments is their lack of realism, which may deter the transfer of technology from the research community to industry. Hence, a challenge is to increase realism while retaining a relatively high likelihood that the technology studied (the treatment) is actually the cause of the observed outcome.

Even though we can increase the realism of experiments, there remain many software engineering phenomena that occur in complex, real-life environments that cannot be studied fully through experiments. We need complementary case studies, particularly when the boundaries between phenomenon and context are not clearly evident. So, while an experiment may choose to deliberately divorce a phenomenon from its context [4], the case study aims to cover contextual conditions. However, embracing more of the context into the focus of study makes it more difficult to identify what affects what. Hence, achieving control is a challenge.

Entity	Sub-entities
Actor	Individual, team, project, organisation, or industry
Technology	Process model, method, technique, tool, or language
Activity	Plan, create, modify, or analyse (a software system) and associated subactivities
Software system	Software systems may be classified along many dimensions: size, complexity, application domain, business/scientific/student project, administrative/embedded/real time, etc.

Table 29.4 Major entities of software development.

The discussion below on realism will be structured around the main entities of software development. The typical situation is that an *actor* applies *technologies* to perform certain *activities* on a (existing or planned) *software system*. Thus, the purpose of a typical software engineering experiment is to compare different technologies in the context of various actors, tasks, and systems. These high-level entities or concepts, with examples of subentities, are listed in table 29.4. One may also envisage collections of entities for each (sub)entity. For example, a software system may consist of requirement specifications, design models, source and executable code, test documents, various kinds of documentation, and so forth. Moreover, the usefulness of a technology for a given activity may depend on characteristics of the software engineers, such as their experience, education, mental ability, personality, motivation, and knowledge of a software system, including its application domain and technological environment.

Increased Representativeness of Subjects (Actors)

In the review reported in [13], only 517 of 5488 subjects were professionals; some were academics, but almost 90 per cent were students. The makeup of the samples makes it very unlikely that the results of these experiments can be generalised to an industry setting:

> Practitioners are understandably sceptical of results acquired from a study of 18-year-old college freshmen. . . finding 100 developers willing to participate in such an experiment is neither cheap nor easy. But even if a researcher has the money, where do they find that many programmers? [12]

At Simula we have focussed on using professionals as subjects (see table 29.2) and we have actually had up to several hundred subjects in one experiment.

A large number of subjects makes it easier to obtain a representative rather than a biased sample of the target population. Only one of 113 experiments reported sampling from a well-defined target population[13]. Moreover, many aspects of the complexity of software engineering, such as differences among subgroups of subjects, only manifest themselves in controlled experiments if they involve a large number of subjects and tasks. For example, in an experiment on pair programming (where programmers work in pairs instead of individually), we wanted to investigate whether there was a difference in the effect of pair programming with respect to seniority among the subjects [8]. We also wanted to test the effect of different levels of system complexity. For the conclusions drawn from a statistical test to be sound, the test must have sufficient power [7]. The three variables pair programming (two levels), control style (two levels), and expertise (three levels) constituted 12 groups in total. The power analysis showed that we needed at least 170 subjects (85 individuals and 85 pairs). The approach of hiring professionals from different companies in different countries to take part made it possible to end up with 99 individuals and 98 pairs. The results showed that the effects of pair programming were dependent on both the seniority of the subjects and the complexity of the systems: The juniors made more correct changes to complex systems when pair programming than when programming alone, whereas the intermediates and seniors spent less time making correct changes on simple system. Hence, the skill or expertise level *relative to the technology being evaluated* must be made explicit, thus indicating the population to which the results apply.

Most experiments in software engineering, including those at Simula, have individuals as the experimental unit. As shown, pairs may also be the unit. Even rarer is to have companies as the unit, although in an experiment on software bidding we used 35 companies as the subjects [11].

Increased Realism of the Studied Technology and Technological Environment

There are two aspects regarding the realism of the technology in an experiment. One is the technology being evaluated; the other one is the technological environment of the experiment. Often the technology being evaluated is developed in a research setting (frequently by the evaluators themselves) and not compared with relevant alternative technology used in the software industry. At Simula, at least in principle, the motivation for studying a given technology should be driven by the needs of industry. The particular technology that we study is related to the topic of investigation. Due to the limited extent of empirical research in software engineering, however, there are many important topics that receive no attention in empirical studies [2, 3]. An overview of topics investigated in software engineering experiments can be found in [13].

There is relatively little reporting of the impact of technology environment on experimental results but it seems clear that using artificial classroom settings without professional development tools can, in many situations, threaten the validity of the results[2]. A review of reported experiments [13] found that the use of computer tools was slightly higher than the use of pen and paper. Only about half of the experiments, however, reported on the use of tools to support assigned tasks. This lack of reporting may be due to a lack of awareness of or interest in this issueÔÃÖs relevance. Nevertheless, the community should recognise the effort and resources needed to set up PC or workstation environments with the right licences, installations, access rights, and so forth, and to familiarise the subjects with the tools. At Simula, we have had the resources to hire professionals to perform the experiments in their own work environment, including development environments. The use of professional development environments has also been supported by the use of the SESE infrastructure described previously.

[2] Note that in smaller experiments, realistic environments may also threaten the validity of the results due to confounding effects associated with the technological environment. For example, in an experiment on Unified Modeling Language [19], the subjects who used a modelling tool spent more time than those who used pen and paper to obtain similar quality. Apparently, those who used the tool spent extra time understanding how to perform tasks with it and getting the syntax correct to avoid error messages.

Increased Realism of Tasks

Activities in software engineering are comprised of tasks, which typically have time limits. Large development tasks can take months, while many maintenance tasks take only a couple of hours. Nevertheless, typical tasks in software engineering experiments are much smaller than typical industrial tasks. In the review reported in [13], the median duration of experiments was 1.0 hour for experiments in which each subject's time was taken and 2.0 hours when only an overall time for the whole experiment was given. About half of the articles mention that the tasks used in the experiments were not representative of industrial tasks and systems with respect to size/duration, complexity, application domain, and so forth.

The community seems to agree that it is a problem that most experiments do not resemble industrial situations, but there is no consensus on what an industrial situation is. There are an endless number of industrial technologies, tasks and systems, so one should be careful to claim that a given technology, task, or system is representative, because it is difficult to specify what it is representative of. In order to meet this challenge, it is necessary to develop well-defined taxonomies that contain representative categories. This can be done by first conducting surveys, logging the activities in certain companies, consulting project information databases, and then analysing the results. Once suitable taxonomies have been developed, experiments can take their samples from the populations that are indicated by the categories of the taxonomies.

Nevertheless, development tasks in industry usually take longer and are more complex than is the case in most experiments (as is the case with technology and systems). Hence, in experiments that have included programming tasks, we have attempted to increase the realism by running the experiments from one day up to two weeks rather than one or two hours [18, 1, 14]. The full-realism experiment on bidding [11] included, by definition, realistic estimation tasks. These tasks can be typical of small, web-based information systems.

Increased Realism of Systems

Most software systems that are used in software engineering experiments are either constructed for the purpose of the experiment or are student projects. In the review reported in [13], only 14 per cent were commercial systems. Accordingly, the systems are generally small and simple. This is also the case for the systems used in the experiments conducted at Simula.

The experiment on pair programming described above demonstrated that system complexity, which is one attribute of a system, might have an effect. In general, however, the research community rarely focuses on what kind of systems are used in the experiments, which may be due to the fact that we do not know how to classify or describe them in a systematic way.

29.6 Increased Control in Case Studies

A case study should be conducted when a "how" or "why" question is being asked about a contemporary set of events investigated within its real-life context [17]. The typical situation in case studies is that there are more variables of interest than data points; hence, there are many possible confounding factors that cannot be controlled for. So, rather than attempting to control for all variables of interest, the context in which the study is conducted should be described in as much detail as possible.

Nevertheless, Simula has launched a new approach in software engineering to achieve more control in case studies. We have found that by running multiple case studies in which certain variables are controlled across the studies, it is more likely that cause-effect relations can be identified. For example, in a study on variability and reproducibility in software engineering [9], the same requirements specification was sent in a call for tenders to 81 software companies. The study was conducted in two parts. In the first part, a randomised controlled field experiment was conducted on the bids from the 35 companies that responded [11]. In the second part, four companies were selected for an in-depth multiple case study in which they developed the system independently, in parallel [9]. The unit of study was thus the company. Figure 29.2 shows the relation between bids and outcomes, which was investigated in the controlled context. Table 29.5 compares the different companies with respect to the project and product quality dimensions.

The system that was referred to previously is a web-based information system to track all the empirical studies conducted by Simula's SE department. The four versions of the system had been running for two years when they needed to be upgraded due to changes in Simula's web environment. Implementing the necessary changes and other corrections was then used as an opportunity to conduct another multiple case study on software maintenance in which six developers from two Eastern European countries implemented the same set of changes on two systems each. The unit of study in this case was thus the individual programmer.

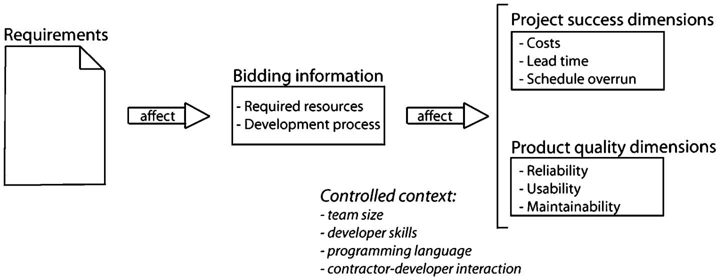


Figure 29.2 Multiple case study with some controlled context.

Dimensions		Company A	Company B	Company C	Company D
Project	Contractor-related costs	90 hours	108 hours	155 hours	85 hours
	Actual lead time	87 days	90 days	79 days	65 days
	Schedule overrun	58%	23%	93%	5%
Product	Reliability	Good	Good	Poor	Fair
	Usability	Good	Fair	Fair	Good
	Maintainability	Good	Poor	Poor	Good

Table 29.5 Quality of project and product.

29.7 Conclusions

The ultimate goal of software engineering research is to support practical software development. Ideally, the research community should provide guidance when industrial users pose the question which method or technology should we use in our context? Of course, this would require sufficient information about a given setting, which in turn requires a wide spectrum of high-quality empirical studies. While the community seems to agree that it is a problem that most experiments do not resemble industrial situations, it is a challenge to define what an industrial situation is. Well-defined, representative taxonomies are needed.

Enabled by, among other things, relatively generous funding for infrastructure and experiment participants, Simula has conducted a large number of industry studies during its first eight years of existence and has, through extensive industry relationships, advanced the state of the art regarding realism in software engineering experiments and control in software engineering case studies. Consequently, we have made some progress in evaluating which software development technologies are useful when, but a joint effort in the software engineering community is needed to further increase realism regarding subjects, technology, tasks, and software systems. We also need more case studies and more control within them. Further progress in this area, however, requires the following: increased competence in conducting empirical studies, improved links between academia and industry, the promotion of common research agendas, and increased funding for empirical studies proportionate to the importance of software systems in society.

References

[1] M. Vokác, W. Tichy, D. I. K. Sjøberg, E. Arisholm, and M. Aldrin. A controlled experiment comparing the maintainability of programs designed with and without design patterns – a replication in a real programming environment. *Empirical Software Engineering*, 9(3):149–195, 2004.

[2] D. I. K. Sjoberg, T. Dyba, and M. Jorgensen. The future of empirical methods in software engineering research. *International Conference on Software Engineering*, pages 358–378. IEEE Computer Society Washington, DC, USA, 2007.

[3] A. Höfer and W. F. Tichy. Status of empirical research in software engineering. *Lecture Notes in Computer Science*, 4336:10, 2007.

[4] J. Hannay and M. Jorgensen. The role of deliberate artificial design elements in software engineering experiments. *Software Engineering, IEEE Transactions on*, 34(2):242–259, 2008.

[5] D. I. K. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanoviċ, E. F. Koren, and M. Vokác. Conducting realistic experiments in software engineering. *ISESE2002 (First International Symposium on Empirical Software Engineering). Nara, Japan*, pages 17–26, 2002.

[6] M. Jørgensen and S. Grimstad. *Software Development Effort Estimation: Demystifying and Improving Expert Estimation. Simula Research Laboratory–By Thinking Constantly About It*, pages 1–2. Springer-Verlag, 2009.

[7] T. Dybå, V. B. Kampenes, and D. I. K. Sjøberg. A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, 48(8):745–755, 2006.

[8] E. Arisholm, H. Gallis, T. Dybå, and D. I. K. Sjøberg. Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Transactions on Software Engineering*, pages 65–86, 2007.

[9] B. Anda, D. I. K. Sjøberg, and A. Mockups. Variability and reproducibility in software engineering: A study of four companies that developed the same system. *IEEE Transactions on Software Engineering*, 2009.

[10] A. Karahasanoviċ, B. Anda, E. Arisholm, S. E. Hove, M. Jørgensen, D. I. K. Sjøberg, and R. Welland. Collecting feedback during software engineering experiments. *Empirical Software Engineering*, 10(2):113–147, 2005.

[11] M. Jorgensen and G. J. Carelius. An empirical study of software project bidding. *IEEE Transactions on Software Engineering*, 30(12):953–969, 2004.

[12] W. Harrison. Skinner wasn't a software engineer. *IEEE Software*, 22(3):5–7, 2005.

[13] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N. K. Liborg, and A. C. Rekdal. A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering*, 31(9):733–753, 2005.

[14] W. J. Dzidek, E. Arisholm, and L. C. Briand. A realistic empirical evaluation of the costs and benefits of UML in software maintenance. *IEEE Transactions on Software Engineering*, pages 407–432, 2008.

[15] E. Arisholm, D. I. K. Sjøberg, G. J. Carelius, and Y. Lindsjørn. A web-based support environment for software engineering experiments. *Nordic Journal of Computing*, 9(3):231–247, 2002.

[16] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen. Action research. *Communications of the ACM*, 42(1):94–97, 1999.

[17] R. K. Yin. *Case study research: Design and methods.* Sage Publications, Thousand Oaks, CA, 3rd edition, 2003.

[18] E. Arisholm and D. I. K. Sjoberg. Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software. *IEEE Transactions on software engineering*, 30(8):521–534, 2004.

[19] B. Anda and D. I. K. Sjøberg. Investigating the role of use cases in the construction of class diagrams. *Empirical Software Engineering*, 10(3):285–309, 2005.

30

A SERIES OF CONTROLLED EXPERIMENTS ON SOFTWARE MAINTENANCE

Erik Arisholm

Erik Arisholm
Simula Research Laboratory

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_30,

PROJECT OVERVIEW

Maintenance of Object-Oriented Software

Software maintenance is both difficult and costly, and software organizations have little tangible evidence with regards to how they can improve their software maintenance practices and precisely what elements comprise the trade-offs, the expected costs, and the benefits of alternative solutions.

Scientific Challenges

The effort required to maintain software systems depends on many interacting factors, such as the software system's structural properties; the use of analysis, design principles and patterns; available documentation of requirements; design and code; tool support; and developer skills. Given the complexities of the problems at hand, empirical studies in realistic yet controlled settings are needed to better understand how such factors drive the costs of software maintenance and the quality of the software artefacts. In the Maintenance of Object-Oriented Software (MOOSE) project at Simula Research Laboratory, we have pursued large-scale, controlled experiments that combine a high degree of realism with the necessary degree of control, so as to obtain credible results that are applicable to the problems faced by the software industry.

Obtained and Expected Results

The studies reported herein contribute to a significantly improved understanding of software maintenance performance by pushing the boundaries of state-of-the-art empirical software engineering research. The results can help software maintenance organizations improve the efficiency of their maintenance processes and the quality of the products being maintained. Specifically, the results provide empirical evidence on trade-offs and expected costs and benefits when applying design principles and patterns, simple task prioritization techniques, the Unified Modelling Language (UML), and pair programming in different maintenance contexts.

A SERIES OF CONTROLLED EXPERIMENTS ON SOFTWARE MAINTENANCE

30.1 Introduction

Software maintenance entails the comprehension of often large, complex systems under constant change, and consumes the majority of software development resources [39]. The ISO 9126 model defines maintainability as *"a set of attributes that bear on the effort needed to make specified modifications."* It can also be viewed as a two-dimensional characteristic involving both the effort expended on implementing changes and the resulting quality of the changes [3]. The effort required to make correct changes to a software system depends on many factors, including characteristics of the software system itself (e.g., code, design, architecture, and documentation), the development environment and tools, the software engineering process used, and human skills and experience. To better understand how maintenance performance is affected by such a complex combinations of factors, empirical studies are needed. As will be demonstrated in this chapter, once a credible body of empirical evidence has been developed, it can be used to propose improvements that can result in large cost savings and, as it turns out, carried out by quite simple means.

The need to address the above issues using controlled experiments stems from the many confounding and uncontrollable factors that could blur the results in an industrial context, making it difficult to establish a causal relationship between independent variables (e.g., design patterns, UML) and dependent variables (e.g., time, correctness). Basili et al. [10] state that "controlled experiments can generate stronger statistical confidence in the conclusions," while Judd et al. [29] write that "we can confidently infer causality from the relationship between two variables only if people have been randomly assigned to the levels of the independent variables."

However, controlled experiments are a compromise, as they can run only for a limited time and necessarily involve smaller tasks and artefacts. Again, this is a well-known drawback: "Unfortunately, since controlled experiments are expensive and difficult to control if the project is too large, the projects studied tend to be small." [9] It therefore raises questions about the extent to which their results can be generalized to realistic tasks, artefacts, and project settings (external validity). In the experiments reported herein, we have pushed the degree of realism as far as possible within the available time and budget constraints. The studies contribute to an improved understanding of software maintenance performance and represent, in several ways, the state of the art within empirical software engineering research. For example, in a controlled experiment on the costs and benefits of the UML, the subjects consisted of senior professional developers who used professional development tools to perform maintenance tasks within a time frame of one to two weeks on a real, non-trivial system that is currently in use. Notwithstanding that single example, more controlled experiments and field studies, such as industrial case studies and surveys, are needed to obtain a comprehensive body of evidence. For example, a case study conducted at Simula on the industrial use of the UML [2] complements and provides additional credibility for the experimental results reported in this paper. During an investigation's early stages, controlled experiments enable the investigators to better understand the issues at stake and the factors to be considered. Furthermore, controlled experiments enable the assessment of whether the results obtained on smaller artefacts and tasks can be considered encouraging, and justify further evaluation of actual use in the field.

In the remainder of this chapter, we focus our attention on controlled experiments that have been conducted in the context of software maintenance by the MOOSE project in the Software Engineering department at Simula and highlight the most important practical and research methodological implications of this work.

30.2 The Experiments

The controlled experiments described in this chapter study combinations of factors that are important drivers of maintenance performance. The experiments are summarized in table 30.1. In the following subsections, we briefly describe the motivation, design decisions, and main results of each experiment. A summary of contributions, in terms of implications of the results on research and practice, is discussed in section 30.6.

Experiment on Control Style

In this experiment, the main research question was to determine how the control style of an object-oriented software system affected the maintainability [8]. According to expert opinion, a *delegated* control (DC) style, typically a result of responsibility-driven design strategies [50], represents object-oriented design at its best, whereas a *centralized* control (CC) style is reminiscent of a procedural solution, or a "bad" object-oriented design [20]. To compare the maintainability of the two control styles, we had previously conducted a small controlled experiment, prior to joining Simula [7]. For the given sample of 36 undergraduate students, the results suggested that the DC style design required significantly more effort to implement the given set of changes than did the alternative, CC style design. This difference in change effort was primarily due to the difference in the effort required to *understand* how to perform the change tasks. We were quite surprised by the results. It was evident that the expert recommendations and the results of our previous experiment ran counter to each other. Thus, we hypothesized that a DC style might provide better software maintainability for an expert, who had the required skills and experience, while a CC style might benefit novices. We wanted to test such hypotheses empirically by

Research question:	Moderator variables:	# subj	Subjects	Treatments	Dependent variable	Tasks	Duration
Control Style [8]	Developer experience	158	BSc students (27) MSc students (32) Juniors (31) Intermediates (32) Seniors (36)	T1: System had a centralized control style T2: System had a delegated control style	Effort Correctness	5 change tasks: 1 pretest task on an ATM + 4 tasks on a coffee machine	12 separate 1-day sessions
Task order [45]	The effect of Control style	125	BSc students (27) MSc students (32) 4th year students (66)	T1: centralized + easy task first T2: delegated + easy task first T3: centralized + hard task first T4: delegated + hard task first	Effort Correctness	5 change tasks: 1 pre-test task on an ATM + 4 tasks on a coffee machine	2 separate 1-day sessions (UiO and NTNU)

Table 30.1 Summary of controlled experiments on software maintenance.

Research question:	Moderator variables:	# subj	Subjects	Treatments	Dependent variable	Tasks	Duration
UML [4]	N/A	98	BSc students UiO: 20 Carleton: 78	T1: System documented with UML T2: System not documented with UML	Effort Correctness Design Quality	6 change tasks: 2 tasks on an ATM and 4 tasks on a coffee-machine	UiO: 1 day Carleton: Four 3-hour labs
UML [23]	N/A	20	Senior professionals	T1: System documented with UML T2: System not documented with UML	Effort Correctness Design Quality	5 change tasks on a realistically sized web-based system	1 to 2 weeks pr. person
Design patterns [44]	Degree of design pattern knowledge (before/after course)	44	39 junior, intermediate or senior consultants 5 students (MSc/PhD)	T1: system with design pattern (PAT), before course T2: PAT, after course T3: system without design |pattern (ALT), before course T4: ALT, after course	Effort Correctness	8 change tasks: 2 change tasks for each of 4 different programs (with or without certain design patterns)	2 days: experiment session 1, a course on design patterns, followed by experiment session 2

Table 30.1 (Continued)

Research question:	Moderator variables:	# subj	Subjects	Treatments	Dependent variable	Tasks	Duration
Pair programming [5]	Developer experience, Control style (system complexity)	295	Juniors (81) Intermediates (102) Seniors (112)	T1: Pair changed a system with a centralized control style T2: Pair changed a system with a delegated control style T3: Individual changed a system with a centralized control style T2: Individual changed a system with a delegated control style	Effort Duration Correctness	5 change tasks: 1 pre-test task on an ATM, solved individually + 4 tasks on a coffee-machine, solved either individually or in pairs	Phase 1: 10 1-day sessions with individual programmers Phase 2: 17 1-day sessions with pair programmers

Table 30.1 (Continued)

including developer experience as a moderator variable in a new experiment. The main research question we attempted to answer in this experiment was the following:

> For the target population of junior, intermediate, and senior software consultants with different levels of education and work experience, which of the two aforementioned control styles is easier to maintain?

A secondary goal of this experiment, which was the first experiment conducted after Simula was established, was to demonstrate the feasibility of conducting more realistic experiments in software engineering research, in particular along the following dimensions:

- **More representative sample of the population.** The target population of this experiment was professional Java consultants, as opposed to students. A total of 99 junior, intermediate, and senior professional consultants from several international consultancies were hired for one day to participate in the experiment. To compare differences between (categories of) professionals and students, 59 students also participated.
- **More realistic tools.** Professional developers use professional programming environments. Traditional pen-and-paper based exercises are hardly realistic. Thus, in this experiment, each subject used an integrated development environment of their own choice, for example, JBuilder, Forte, Visual Age, Visual J++, and Visual Café.
- **More realistic experiment environment.** The classroom environment of most previous experiments was replaced by the office environment in which each developer would normally work. Thus, they had access to printers, libraries and so forth as they would for any other project on which they might be working. The students were located in one of their campus computer labs.

The motivation was to increase the external validity of results. Up to that point, most experiments in software engineering had been performed with students, who often did use pen-and-paper experimental systems and tasks. However, increased realism also meant increased logistical complexity of the experiment conduct. Thus, to support the logistics of the experiment, the web-based Simula Experiment Support Environment (SESE) [6] was developed. With SESE, the subjects could answer questionnaires, download code and documents, and upload task solutions while being monitored by the researchers through the web-based interface.

The results showed that the developers with the most skills, in particular the senior consultants, required less time to maintain software with a DC style than with a CC style, without reductions in functional correctness. However, more novice developers, in particular the undergraduate students and junior consultants, experienced serious problems understanding a DC style and performed far better with a CC style, both in terms of time spent and the functional correctness of the delivered tasks. This interaction effect clearly demonstrates the importance of including developer experience as a moderator variable. The results are depicted in figure 30.1.

The experiment offers guidance on how to delegate responsibilities among classes in an object-oriented system in order to improve its maintainability. For example, in

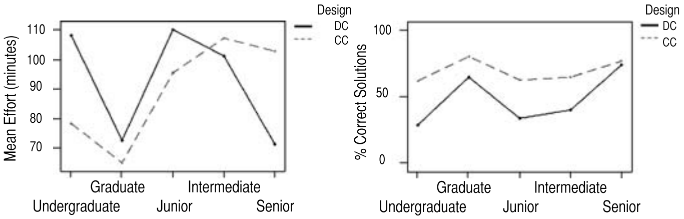


Figure 30.1 Effect of DC versus CC control systems on effort and correctness for categories of developer.

a use-case driven analysis process, one would typically assign one control-object for each use case, and this control-object is responsible for coordinating the flow of events between the user interface and the entity objects that participate in the use case. But how much of the business logic should be handled by the control-object and how much should be delegated to the entity objects? Proponents of responsibility-driven design would argue that the responsibility of the control-object should be small and should delegate most of the business logic responsibilities to the entity objects, according to principles such as "information expert" [40]. However, this experiment shows that such a strategy may result in designs that are difficult to understand for inexperienced maintainers. Assuming that it is not only highly skilled experts who will maintain an object-oriented system, a viable conclusion drawn from this controlled experiment is that a design with a CC style may be easier to maintain than a design with a DC style.

Experiment on Task Order

In many cases, there are few constraints on how to break down and arrange maintenance tasks, in which case, one can choose freely among alternative strategies to prioritize or sequence the tasks. If we could determine any indicators showing that the order of change tasks can affect the future maintainability of the system, the result could represent a high return on investment for software companies, because little effort is required to rearrange task order. Thus, we wanted to assess the effects of ordering maintenance tasks with respect to difficulty level. Specifically, the study reported in [45] compared how the change effort and correctness is affected by starting with the easiest change task and progressively performing the more difficult tasks (Easy-first), versus starting with the most difficult change task and progressively performing the easier tasks (Hard-first).

Hence, the experiment attempted to assess two competing theories:

1. By starting with the easy maintenance tasks first, the learning curve will not be very steep, thus enabling the maintainers to obtain a progressively better overview of the software system before having to perform more difficult tasks. In this way, the maintainer is less likely to devise suboptimal solutions when per-

forming the difficult tasks. This is closely related to what is defined as a bottom-up strategy regarding program comprehension, in which programmers look for small, recognizable patterns in the code and gradually increase their knowledge of the system [33].

2. By starting with the difficult maintenance tasks first, the learning curve will be steep, as the programmer must obtain a more complete overview of the system before being able to make changes. However, due to the better overview, the maintainer might be less likely to devise suboptimal task solutions. This is related to a top-down strategy regarding program comprehension, in which the programmer forms hypotheses and refinements of hypotheses about the system that are confirmed or refuted by items of the code itself [16].

In both cases, the experimental tasks were performed on two alternative control styles of a Java system (centralized versus delegated) to assess whether the choice of the design strategy moderates the effects of task order on effort and correctness.

A unique feature of this experiment was in the way the subjects were assigned to the treatments. Data from the 59 third to fith year students from UiO who participated in the previous experiment on the effect of control style [8] were reused, as they had performed the tasks in an Easy-first task order. The second phase of the experiment was conducted with 66 mostly fourth-year students at NTNU. The NTNU students followed the same experimental procedures and used the same materials as had been used in the first experiment, except they started with the most difficult task (randomly assigned to either the CC or DC design) and progressively performed the easier tasks (Hard-first). It can thus be classified as a differentiated replication [42] of the first experiment. Together, the two experiments formed one quasi-experiment, since there was a non-random assignment to the treatment groups. To adjust for individual skill differences between the treatment groups, all subjects in both of the controlled experiments performed the same pretest programming task. The results of the pretest were then used in an analysis of covariance (ANCOVA) model of the effect of task order on maintainability [22]. Although this is a common approach for analyzing quasi-experiments [22], using a pretest in this way is surprisingly uncommon in software engineering experiments.

Figure 30.2 provides an overview of the experiment design that shows the differences between the UiO and NTNU experiments. The first three steps were the same for both experiments. However, for tasks C1–C3, the subjects carried out four variants of the treatment with variation in two dimensions: Task order (Hard-first vs. Easy-first) and Control style (DC vs. CC). The last step (task C4) was the same for both experiments, but with two variations (DC and CC). The C4 task occupied the same place in the sequence for both the UiO and NTNU experiments, functioning as a benchmark task for which we tested whether effort or correctness for that task was affected by the preceding task order.

The results showed that the time spent on performing the benchmark task was not affected significantly by the task order of the preceding tasks, regardless of the control style (centralized versus delegated). However, the correctness of the benchmark task was significantly higher when the task order of the preceding change tasks was Easy-first compared with Hard-first, again regardless of design. The estimated

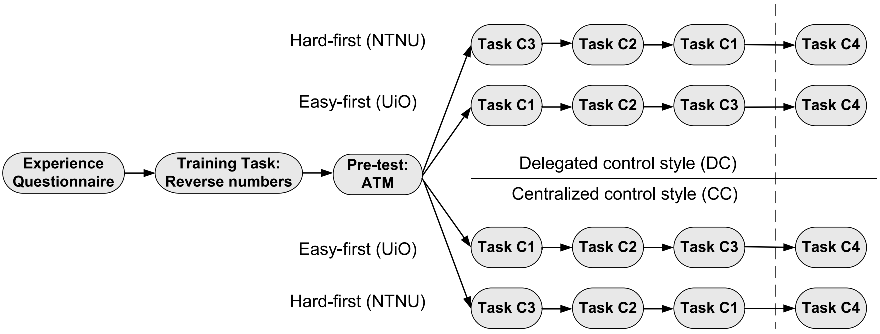


Figure 30.2 Experiment design for the task order experiment.

difference in correctness was substantial: With the Easy-first task order, 46 per cent of the subjects were able to correctly perform the benchmark task. With the Hard-first task order, only 12 per cent of the subjects were able to correctly perform the benchmark task.

30.3 Experiments on UML

Software maintenance is often performed by individuals who were not involved in the original design of the system being changed. This is why documenting software specifications and designs often has been advocated as a necessity to help software engineers remain in intellectual control while changing complex systems [14, 18]. Indeed, it is expected that many aspects of a software system must be understood in order to properly change or modify it, including its functionality, architecture, and a myriad of design details.

However, documentation entails a significant cost and must be maintained along with the software system it describes. The issue that then arises is what content and level of detail are required for efficient software maintenance[15]. The proponents of agile methods usually advocate keeping documentation to a minimum and focusing on test cases as a source of system requirements [11]. By contrast, proponents of model-driven development view software development as a series of modelling steps, in which each step refines the models of the previous step [31]. The transition from analysis to high-level design to low-level design to code is supported by tools facilitating and automating parts of the work. Modelling is seen as a way to better handle the growing complexity of software development by helping engineers to work at higher levels of abstraction. Model-driven development is supported by UML [14], an evolving standard that is now widespread across the software industry. However, despite its growing popularity, there is little reported evaluation of the use of UML-based development methods [2], and many perceive the documentation of analysis and design models in UML to be a wasteful activity [11]. Hence, such practices are viewed as difficult to apply in development projects where resources and time are tight.

It is then important, if not crucial, to investigate whether the use of UML documentation can make a practical and significant difference that would justify the costs to create it. As a first step in this direction, we conducted two controlled experiments that investigated the impact of UML documentation on software maintenance. The first experiment was with students who had appropriate training in UML from the University of Oslo and Carleton University in Canada. The systems and tasks were, with some exceptions, identical to those of the control-style and task order experiments described in the previous sections. The second experiment built on the expertise and results of the first experiment, but the second experiment's primary strength was the level of realism: it involved 20 professional developers (intermediate- to senior-level consultants) individually performing the same five maintenance tasks to the same real, non-trivial system wherein ten of the developers worked with a UML-supported development environment and UML documentation, whereas the other ten developers used the same tools but had no UML documentation to read or update. The developers took one to two weeks to implement the change-tasks.

The results of the two experiments were, for the most part, consistent. Both students and professionals benefited greatly from having UML documentation, in terms of their ability to perform correct change tasks on both relatively small and large systems. Overall, there was about a 50 per cent increase in the proportion of correct tasks for programmers who had UML documentation, which is substantial by any standard. UML also helped the developers in not violating the original design, for example, by using existing methods when appropriate, as opposed to duplicating code or writing redundant code. Qualitative results from the experiments suggested that a major reason for these benefits was the utilization of UML sequence diagrams, which helped the developers to better understand the control flow and delegation of responsibilities among the participating objects in the systems. Such information made it much easier to make correct changes. However, updating the documentation resulted in a time overhead of between 10 and 20 per cent, depending on the system and task. Qualitative results suggested that a major reason for this was that the UML tools had very poor support for efficient updating of sequence diagrams in particular.

30.4 Experiment on Design Patterns

Software design patterns seek to package proven solutions to design problems in a form that makes it possible to find, adapt and reuse them. Design patterns have become quite popular [19, 27]. In addition to making design knowledge available to both junior and more experienced developers, it is claimed that design patterns define a common terminology that can be used to document the design. According to the classic book by Alexander [1], individual patterns can be combined into a language that guides the designer. This should simplify communication of the underlying design and assumptions held by the original designers to maintainers of the software. Design patterns tend to provide solutions that are more complete and general than merely solving the immediate problem at hand. The same property may, however, introduce unneeded complexity.

To support the industrial use of design patterns, this research investigated when and how using patterns could be beneficial, and whether some patterns are more difficult to use than others. The experiment was a replication of an earlier controlled experiment on design patterns in maintenance [38], with major extensions. Experimental realism was increased by using a real programming environment instead of an academic pen-and-paper exercise, and paid professionals from multiple major consultancies were used as subjects. Measurements of elapsed time and correctness were analyzed using regression models and an estimation method that took into account the correlations present in the raw data. Together with online logging of the subjects' work, this made possible a better qualitative understanding of the results. The results indicated quite strongly that some patterns are much easier to understand and use than others. In particular, the Visitor pattern caused much confusion. Conversely, the patterns Observer and, to a certain extent, Decorator were grasped and used intuitively, even by subjects with little or no knowledge of patterns.

Thus, the implication is that design patterns are neither universally good nor bad, but must be used in a way that matches the problem and the users. When approaching a program with documented Design Patterns, even basic training can improve both the speed and quality of maintenance activities.

30.5 Experiment on Pair Programming

The concepts underlying pair programming (PP) are not new [21, 23, 24, 46], but PP itself has only recently attracted significant attention and interest within the software industry and academia. Much of the focus on PP is due to the introduction of extreme programming (XP), in which PP is one of twelve key practices [12, 11].

Basically, in PP, two programmers work on the same task using one computer and one keyboard [12, 11, 48, 47]. There are two distinct roles that contribute to a synergy of the individuals in the pair: (1) a driver, who types at the keyboard and focuses on the details of the coding, and (2) a navigator, who actively observes the work of the driver, looking for tactical and strategic defects, thinking of alternatives, writing down "things-to-do", and looking up references. In addition to coding, PP also involves other phases of the software development process, such as design and testing.

Several earlier controlled experiments have concluded that PP offers many benefits over individual programming, including significant improvements in functional correctness and various other measures of quality of the programs being developed, reduced duration (a measure of time to market), and only minor additional overhead in terms of total programmer hours (a measure of cost or effort) [34, 35, 37, 48, 47]. However, most of the existing studies cannot be compared directly, due to differences in sample populations (e.g., students or professionals), study settings (e.g., amount of training in PP), lack of power (e.g., few subjects), and different ways of treating the dependent variables (e.g., how correctness was measured and whether measures of development time also included rework) [26, 35, 36]. Furthermore, a common feature of the existing studies is that they have not accounted for the moderating effect of the complexity of the programming tasks, which in turn may depend on the complexity of the system being developed or maintained, and the expertise of the programmers. In light of existing research in software engineering [8, 28, 41] and social psychology [13, 17, 25, 30, 51], we expected that system complexity and programmer expertise would have a significant impact on when and how PP is beneficial compared with individual programming [26]. To investigate these issues empirically, we conducted an experiment that addressed the following research question:

> What is the effect regarding duration, effort, and correctness of pair programming for various levels of system complexity and programmer expertise when performing change tasks?

The dependent variables *duration*, *effort*, and *correctness* represent, respectively, one dimension of more general concepts such as time to market, costs, and quality.

Previous experiments on PP have been conducted with students [35, 36, 48] or with only a few professionals [34, 37] (both experiments with five pairs and five individuals). To have sufficient power to investigate our research question and to obtain a relatively representative sample of Java programmers, we conducted a two-phase experiment with a total sample of 295 Java consultants (98 pairs and 99 individuals) from 29 consultancies in Norway, Sweden, and the UK, including international companies such as Accenture, Cap Gemini, CIBER, Oracle, Steria, and TietoEnator. The first phase of the experiment was conducted on individual developers in 2001 [8]; the second phase was conducted on developer pairs in the second half of 2004 and the first half of 2005. First, all subjects performed a pretest task, the results of which were used to adjust for skill differences between these two groups. Subsequently, the subjects performed change tasks on two alternative Java systems based on a CC or a DC style, respectively [49].

The results of this experiment are depicted in figure 30.3. It did not support the hypotheses that pair programming in general reduces the time required to solve the tasks correctly or increases the proportion of correct solutions. On the other hand, there was a significant 84 per cent increase in effort to perform the tasks correctly. However, on the more complex system (with a DC style), the pair programmers showed a 48 per cent increase in the proportion of correct solutions, but no significant difference in the time taken to solve the tasks correctly. For the simpler system, there was a 20 per cent decrease in time taken but no significant difference in correctness. However, the moderating effect of system complexity depended on the programmer expertise of the subjects. The observed benefits of pair programming in terms of correctness on the complex system applied mainly to juniors (149 per cent increase), whereas the reductions in duration to perform the tasks correctly on the simple system applied mainly to intermediates (39 per cent decrease) and seniors (23 per cent decrease).

30.6 Contributions

The experiments presented here have contributed to a better understanding of how software design principles, design documentation, and process-related factors, such

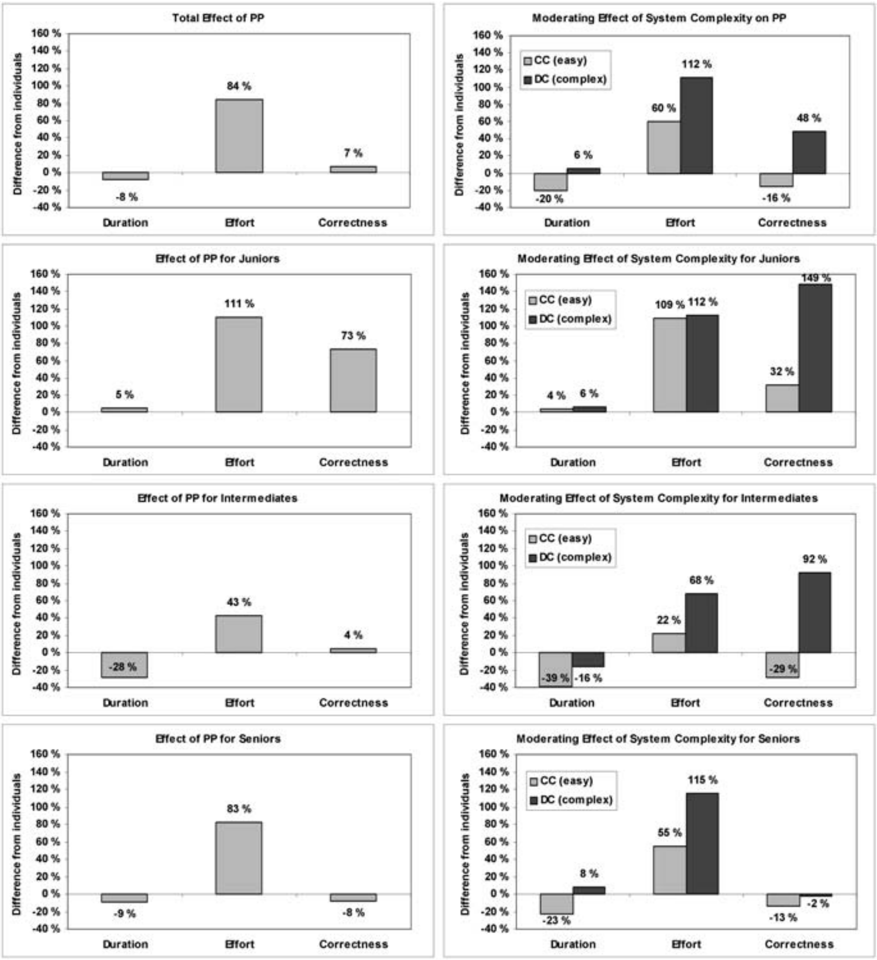


Figure 30.3 Summary of results of the pair programming experiment.

as task order and pair programming affect software maintenance performance and provide practical, industry-relevant advice on how to improve performance. The studies also contribute to the research methodology.

Practical implications

Our two experiments on fundamental object-oriented analysis and design principles (control style and design patterns, respectively) exhibit some important similarities and lessons learned. For example, most experienced software designers would probably agree that a DC style is more "elegant", and a better object-oriented representation of the problem to be solved, than is a CC style. Similarly, as design patterns become more well-structured, general, reusable, and offer documented solutions to common design problems, then their benefits will come to be indisputable. Clearly, the DC style and the appropriate use of design patterns result in solutions that are more changeable in the purely technical sense. Solutions will be more modular and well-structured, resulting in less coupling and more cohesive classes. Consequently, once a maintainer understands such designs, effort will be saved, because one can more easily reuse existing design elements, and the resulting changes will probably be smaller and with fewer negative side effects.

However, as was clearly demonstrated by these experiments, the degree of maintainability of a software application depends not only on attributes of the software itself, but also on certain cognitive attributes of the particular developer whose task it is to maintain it. The design pattern experiment showed that some patterns are difficult to understand and apply even for very experienced developers, even when they have recently received comprehensive training on the given pattern. Furthermore, from a practical standpoint, we cannot assume that all programmers in industry will be equipped with sufficient training and skills. And even with extensive training, the cognitive abilities of programmers will vary considerably. The admittedly "sensible" principles, such as responsibility-driven design and the use of design patterns come with additional cognitive complexity, resulting in higher costs related to learning and comprehension, and using them may cause a higher error rate for a large portion of the current maintenance personnel. This aspect seems to be underestimated by expert designers. Furthermore, researchers should be aware of such trade-offs, for example, when proposing cohesion as an indicator for changeability or when proposing design-pattern-based refactoring approaches and tools (cf. [30, 32]).

Fortunately, our experiments have also shown that there are several ways some of the negative aspects, with regards to comprehension costs, can be resolved. First, whenever possible, inexperienced maintainers should consider starting with simple tasks, as this will ease the system learning curve considerably and mitigate undesirable side-effects; our experiment on task order indicated that this strategy may actually result in more maintainable software overall. This is closely related to the earlier mentioned bottom-up strategy regarding program comprehension, in which programmers look for small, recognizable patterns in the code and gradually increase their knowledge of the system [33].

Another way to ease the cognitive load of comprehending complex object-oriented software is to provide UML models of the system. Our novel experiments on the costs and benefits of UML show that with the aid of proper models, even inexperienced developers will be able to fully understand and change relatively complex object-oriented software. For example, the results showed that with proper documentation, the proportion of correct solutions for change tasks on a system with a DC style increased substantially, to the extent that the added cognitive complexity of such designs (as compared with CC styles) no longer represented a barrier to problem-solving. It appears that the UML sequence diagrams are particularly useful in helping to understand an object-oriented program. One possible explanation of the results is discussed in [43]: Delocalized plans need to be documented explicitly in higher-level representations of the code to aid in program understanding and maintenance. The authors question whether it is even reasonable to look at the details of program code in order to understand a (delocalized) program [43].

However, the development and maintenance of UML models also come with associated costs. The overhead of keeping the models up-to-date with the code represented an increase of 10 to 20 per cent of the total change effort. Given the 50 per cent increase in correctness and the increase in design quality, this overhead seems a small price to pay. However, qualitative results from the experiments also clearly indicated that the current UML tools are far from optimal. The user friendliness and functionality of the current tools clearly must be improved to facilitate industry-wide adaptation of UML. Our experiments have provided several, empirically based recommendations for improving the tools [23].

A third way to increase maintenance performance is to use pair programming on complex tasks, as is clearly demonstrated in our comprehensive experiment on pair programming. On complex tasks and with inexperienced maintainers, the benefits of pairing up might be worth the added cost, for example, as demonstrated by the 149 per cent increase in correctness for junior developers on the DC style design. In fact, our results suggest that pairs of juniors are able to deliver the same proportion of correct solutions as that rendered by individual senior developers. When seniors are unavailable for performing complex maintenance tasks, pairs of juniors present a viable option.

30.7 Research Methodological Contributions

One important research methodological contribution of the reported experiments is how they explicitly control for moderator variables, such as task complexity, developer experience, and degree of training in the experiment designs. This has been a very uncommon control aspect in software engineering experiments. The results also show how essential it is to consider such moderating factors. There is no one universal answer, for example, to the *effect* of: control style (dependent on developer experience), design patterns (dependent on the level of training), or pair programming (dependent on both developer experience and task or system complexity).

Another contribution of the control-style, task order, and pair programming experiments is that they demonstrate how an individual pretest can be used to adjust for non-equivalent group designs. The task order and pair programming experiments actually reused observations from the control-style experiment. Due to the high cost of recruiting subjects, planning, and conducting these experiments, it makes practical sense to reuse data. The results of the pretest can be used as a covariate in an ANCOVA model to adjust for between-group differences, due to the non-random assignment to the treatment groups. This method enabled us to combine experimental results to study combinations of factors in a cost-effective way.

Finally, as part of this research, the web-based SESE [6] was developed. SESE supports the logistics of running large-scale, more realistic experiments, and it has shown to be particularly useful in situations in which the subjects are located in several development sites or even in different countries, for example, as was the case in the pair programming experiment. With SESE, the subjects can answer questionnaires, download code and documents, and upload task solutions via the web. Also, SESE helps to ensure that the subjects perform the sequence of experimental tasks in accordance with the defined procedures and provides monitoring functionality for the researcher. Furthermore, experiments can easily be replicated, and experimental designs and materials can be reused to answer new research questions.

30.8 Conclusions

We must perform empirical studies to better understand how the maintenance process, software design principles, properties of the software artefacts, including documentation, human and organizational factors, and supporting tools affect software maintainability. Software maintenance is costly, and even small improvements can offer substantial cost savings. This paper has described some controlled experiments that have been performed on software maintenance in the MOOSE research project in the SE department at Simula. In many ways, these experiments have pushed the boundaries of the state of the art, in terms of the research questions being posed, degree of realism, scale, rigor of experiment conduct, analyses, and the practical implications of the results.

First, we have clearly identified the trade-offs between improving the technical changeability of software, as with the use of design patterns and responsibility-driven design, and the consequential added comprehension costs during software maintenance. Furthermore, we have demonstrated how maintenance costs can be reduced by means of simple ways of scheduling the order of maintenance tasks, the use of UML, and using pair programming on complex tasks, all of which contribute to better comprehensibility of the maintained software.

Research methodological contributions include demonstrating the importance of designing experiments, so that moderator variables such as developer experience and system complexity can be studied; demonstrating how systems, tasks, and subjects of individual experiments can be effectively reused to form even larger scale quasi-experiments; and a tool that supports the logistics of conducting large-scale, controlled experiments.

References

[1] C. Alexander. *A Pattern Language: Towns, Buildings, Construction.* Oxford University Press Inc, 1978.

[2] B. Anda, K. Hansen, I. Gullesen, and H. Thorsen. Experiences from using a uml-based development method in a large safety-critical project. *Empirical Software Engineering,* 11(4):555–581, 2006.

[3] E. Arisholm. Empirical assessment of changeability in object-oriented software. Master's thesis, University of Oslo, 2001.

[4] E. Arisholm, L. C. L. Briand, S. E. Hove, and Y. Labiche. The impact of uml documentation on software maintenance: An experimental evaluation. *IEEE Transactions on Software Engineering*, 32(6):365–381, 2006.

[5] E. Arisholm, H. Gallis, T. Dybå, and D. I. Sjøberg. Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Transactions on Software Engineering*, 33(2):65–86, feb 2007.

[6] E. Arisholm, D. Sjøberg, G. J. Carelius, and Y. Lindsjørn. A web-based support environment for software engineering experiments. *Nordic Journal of Computing*, 9(4):231–247, 2002.

[7] E. Arisholm, D. Sjøberg, and M. Jørgensen. Assessing the changeability of two object-oriented design alternatives - a controlled experiment. *Empirical Software Engineering*, 6(3):231–277, 2001.

[8] E. Arisholm and D. I. K. Sjøberg. Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software. *IEEE Transactions on Software Engineering*, 30(8):521–534, aug 2004.

[9] V. Basili. The role of experimentation in software engineering: past, current, and future. *proc. IEEE International Conference on Software Engineering*, pages 442–449, 1996.

[10] V. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4):456–473, July/August 1999.

[11] K. Beck. *Extreme Programming Explained.* Addison Wesley, 2001.

[12] K. Beck. Embrace change with extreme programming. *IEEE Computer*, 32(10):70–77, 1999.

[13] C. F. Bond and L. J. Titus. Social facilitation: A meta-analysis of 241 studies. *Psychological Bulletin*, 94(2):265–292, 1983.

[14] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide.* Addison Wesley, 1999.

[15] L. Briand. Software documentation: How much is enough? *proc. IEEE European Conference on Software Maintenance and Reengineering*, pages 13–15, 2003.

[16] R. Brooks. Towards a theory of the cognitive processes in computer programming. *Int. Journal on Man-Machine Studies*, 9:737–751, 1977.

[17] R. Brown. *Group Processes.* Blackwell Publishing, 2 edition, 2000.

[18] B. Bruegge and A. Dutoit. *Object-Oriented Software Engineering Using UML, Patterns, and Java.* Prentice Hall, 2nd edition, 2004.

[19] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-Oriented Software Architecture.* Wiley, 1996.

[20] A. Cockburn. The coffee machine design problem: Part 1 & 2. *C/C++ User's Journal*, (May/June), 1998.

[21] L. L. Constantine. *Constantine on Peopleware.* Prentice-Hall, 1995.

[22] T. D. Cook and D. Campbell. *Quasi-Experimentation - Design & Analysis Issues for Field Settings.* Houghton Mifflin Company, 1979.

[23] J. Dzidek, E. Arisholm, and L. C. Briand. A realistic empirical evaluation of the costs and benefits of uml in software maintenance. *IEEE Transaction on Software Engineering*, 34(3):407–432, 2008.

[24] N. V. Flor and E. L. Hutchins. Analyzing distributed cognition in software teams: A case study of team programming during perfective software maintenance. *proc. Fourth Workshop on Empirical Studies of Programmers*, pages 36–64, December 7-9 1991.

[25] D. R. Forsyth. *Group Dynamics.* Wadsworth Publishing Company, 3 edition, 1999.

[26] H. Gallis, E. Arisholm, and T. Dybå. An initial framework for research on pair programming. *proc. 2003 ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2003*, pages 132–142, September 30 - October 1 2003.

[27] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of reusable object-oriented software.* Addison-Wesley, 1995.

[28] T. Hærem. Task complexity and expertise as determinants of task perceptions and performance. Master's thesis, Norwegian School of Management BI, 2002.

[29] C. Judd, E. Kidder, and L. Smith. *Research methods in social relations.* Holt, Rinehart and Winston, Inc., 6th edition, 1991.

[30] S. J. Karau and K. D. Williams. Social loafing: A meta-analytic review and theoretical integration. *Journal of Personality and Social Psychology,* 65(4):681–706, 1993.

[31] A. Kleppe, J. Warmer, and W. Bast. *MDA Explained - The Model Driven Architecture: Practice and Promise.* Addison-Wesley, 2003.

[32] A. Lakhotia and J.-C. Deprez. Restructuring functions with low cohesion. *proc. IEEE Working Conference on Reverse Engineering (WCRE)*, pages 36–46, October 6–8 1999.

[33] S. Letovsky and E. Soloway. Delocalized plans and program comprehension. *IEEE Software,* 3(3):41–49, 1986.

[34] K. M. Lui and K. C. Chan. *When Does a Pair Outperform Two Individuals? Extreme Programming and Agile Processes in Software Engineering,* volume 2675/2003, pages 225–233. Springer Lecture Notes on Computer Science (LNCS), 2003.

[35] M. M. Müller. Two controlled experiments concerning the comparison of pair programming to peer review. *Journal of Systems and Software,* 78(2):166–179, nov 2005.

[36] J. Nawrocki and A. Wojciechowski. Experimental evaluation of pair programming. *proc. European Software Control and Metrics (Escom),* April 2-4 2001.

[37] J. T. Nosek. The case for collaborative programming. *Communications of the ACM,* 41(3):105–108, 1998.

[38] L. Prechelt, B. Unger, W. F. Tichy, P. Brossler, and L. G. Votta. A controlled experiment in maintenance comparing design patterns to simpler solutions. *IEEE Transactions on Software Engineering,* 27(12):1134–1144, 2001.

[39] R. Pressman. *Software Engineering - A Practitioner's Approach.* McGraw Hill, 7th edition, 2005.

[40] R. Pressmann. *Software Engineering. A Practitioner's Approach.* McGraw-Hill, 1997.

[41] S. D. Sheetz. Identifying the difficulties of object-oriented development. *Journal of Systems and Software*, 64(1):23–36, 2002.

[42] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, A. Kampenes, A. Karahasanovic, K. L. N, and A. C. Rekdal. A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering*, 31(9):1–21, 2005.

[43] E. Soloway, R. Lampert, S. Letowski, D. Littman, and J. Pinto. Designing documentation to compensate for delocalized plans. *Communications of the ACM*, 31(11):1259–1267, 1988.

[44] M. Vokác, W. Tichy, D. I. K. Sjøberg, E. Arisholm, and M. Aldrin. A controlled experiment comparing the maintainability of programs designed with and without design patterns – a replication in a real programming environment,. *Empirical Software Engineering*, 9(3):149–195, 2004.

[45] A. I. Wang and E. Arisholm. The effect of task order on the maintainability of object-oriented software. *Information and Software Technology*, 51(3):293–305, 2009.

[46] G. M. Weinberg. *The Psychology of Computer Programming*. Van Nostrand Reinhold Company, 1971.

[47] L. A. Williams. The collaborative software process. Master's thesis, University of Utah, 2000.

[48] L. Williams, R. R. Kessler, W. Cunningham, and R. Jeffries. Strengthening the case for pair programming. *IEEE Software*, 17(4):19–25, 2000.

[49] R. Wirfs-Brock. Characterizing your application's control style. *Report on Object Analysis and Design*, 1(3), Sept/Oct 1994.

[50] R. Wirfs-Brock and B. Wilkerson. Object-oriented design: A responsibility driven approach. *SIGPLAN Notices*, 24(10):71–75, 1989.

[51] R. B. Zajonc. Social facilitation. *Science*, (149):269–274, 1965.

PART III

RESEARCH EDUCATION

Kristin Vinje, Director of the Simula School of Research and Innovation, and Are Magnus Bruaset, Assistant Director of the Simula School of Research and Innovation.

31

EDUCATING RESEARCHERS — A VIRTUE OF NECESSITY

Are Magnus Bruaset and Kristin Vinje

Abstract The training of next-generation researchers is important to academia, to industry, and to society. Therefore, research education is one of Simula's three main goals, alongside conducting basic research of outstanding quality, and bringing the research results into use in both business and society.

In 2007, the education of master and PhD students and the training of postdoctoral fellows at Simula became centred around the Simula School of Research and Innovation. This unit represents a novel approach, in that it organises the educational programme within a limited company, co-owned by Simula Research Laboratory and several industrial companies.

In this chapter, we explain the role that research education plays in Simula, highlight the results obtained, and draw an outline of the future ambitions for Simula's research education.

Why is Research Education Important for Simula?

In a recent interview[1], Vice President Ingolf Søreide of StatoilHydro's research centre in Trondheim stated that the company is "simply making a virtue of necessity" when it is heavily involved in Simula's educational programme. As an internation-

Are Magnus Bruaset · Kristin Vinje
Simula Research Laboratory

Are Magnus Bruaset
Department of Informatics, University of Oslo, Norway

[1] See the interview on page 495.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_31,

ally leading oil and gas company, StatoilHydro's rationale is to ensure that the best students are properly trained as researchers and thereby become the future experts needed to help the company excel in exploration and production. This argument extends directly to other industrial branches that also are highly dependent on technology and, further, on to academic research. There is a wide industrial demand for scientifically strong candidates with a relevant background, but the same type of candidates are also needed to enrich and develop the basic research community.

Since the early 19th century, when Wilhelm and Alexander Humboldt established their university in Berlin, research and teaching have been intimately connected. The Humboldt educational system has been a higher-education model for many countries, including Norway. In this tradition, it is only natural that education be one of the pillars of Simula. However, given the ambitious research profile of the laboratory, its educational efforts have always been concentrated on the master, doctoral, and postdoctoral levels; that is, the focus is on education *for research*, targeting both academia and knowledge-driven industry. This focus is also firmly anchored in Simula's strategy for 2007–2015; it states that the goal is

> to increase the production of MSc and PhD candidates, and... continue to foster independent and cooperative researchers with high scientific and ethical standards.

To reach this goal, Simula is to "deliver high-quality graduate education in partnership with Norwegian universities." In most cases, the students affiliated with Simula receive their academic degrees from the University of Oslo. Although there is a special emphasis on graduate education, and personnel from Simula also contribute to the teaching of undergraduate courses at UiO. As one particular reward, this type of teaching helps to funnel strong candidates into projects of interest to Simula.

Research education is important to Simula for several reasons. Locally, it serves the laboratory's research goals both by offering the valuable work capacity of highly competent PhD students and postdoctoral fellows, and by supplying a steady stream of postdoctoral candidates capable of filling senior positions. In a wider context, Simula is expected to contribute to society by providing excellent training for researchers in relevant fields of information technology. These researchers are in demand in industry as well as in the public sector. This expectation has defined research education as one of the three main axes of Simula, alongside basic research of outstanding quality and the transfer of knowledge from research to practical use. Since the reorganisation of Simula into a corporate structure in January 2008, this role has been clearly demonstrated in that *Research Education* has become one of the three corporate units, complemented by Basic Research and Research Applications.

Room for improvement. Although the education of PhD students has always been a priority at Simula, the evaluation[2] of the laboratory in 2004[1], pointed out that there is room for improvement. The evaluation committee recommended that Simula should raise its postgraduate student targets, as the ratio of the number of PhD students to the number of staff members was low, compared with other world-class

[2] See page 67 for information about the evaluation.

academic institutions. When seeking to improve this ratio, the greatest challenge is securing the necessary funding, and this dilemma was also recognised by the evaluation committee.

In contrast to many other countries, PhD students in Norway are employed at a university, at a research institute, or at a company. The generally high level of salaries in this country has led to high salaries for the PhD students; moreover, the costs of office space and infrastructure, and supervision are also high. As a result, the PhD education in Norway is among the most expensive worldwide. Based on the Research Council of Norways's official rate for the annual cost of a PhD student, the total cost of a three-year doctoral education is 2.4 million NOK. In comparison, Simula's corporate budget for 2009 is about 100 million NOK.

In summary, an increase of the volume of PhD students at Simula is primarily a matter of increasing our level of large-scale funding, and is currently not limited by the available capacity for supervision. Consequently, the fulfilment of Simula's ambitious goals for research education requires either a change in the policy governing the allocation of public funds for doctoral education, or an increase of industrial collaboration that involves positions for PhD students. As presented in section 1, Simula has recently been highly successful in attracting industrial funding for educational positions. Also, support from public agencies has increased, but to a lesser degree than what is needed to realise the ambitious goal of achieving 15 awarded PhD degrees annually.

How Does Simula Educate Next-generation Researchers?

The three research departments at Simula have always been heavily involved in the education of master and doctoral students, even in their origin as research groups at UiO prior to Simula's formation in 2001. However, inspired by the criticism raised in the evaluation in 2004, regarding the low number of PhD students per staff member, Simula has taken important steps to improve and extend its educational efforts. In particular, the establishment of the Simula School of Research and Innovation AS has opened new possibilities that involve long-term educational partnerships with leading Norwegian industrial companies. This unit represents a novel approach in that it organises the educational programme into a limited company. Simula Research Laboratory AS (56%) is the majority owner of SSRI. In addition, our main industrial partners, StatoilHydro (21%) and Telenor (7%), have entered as active owners. The remaining owners are the Municipality of Bærum (14%), SINTEF (1%), and the Norwegian Computing Center (1%).

There is a strong communion between Simula and SSRI. Although PhD students and postdoctoral fellows are employees of SSRI, they are also fully accredited as project members in the relevant basic research departments. Physically, the students, fellows and staff are co-located. The basic research departments are responsible for scientific supervision and contribute financially by using the share of their funding that is reserved for PhD and postdoctoral positions. On the other hand, SSRI plays an important role in quality assurance of the student-supervisor relationship, in industrial networking, and in the development and implementation of courses that extend the traditional university portfolio. By collecting all educational activities into SSRI, Simula has also increased the visibility of these efforts, both internally and in the political context.

Establishing Simula School of Research and Innovation. The history of Simula's research groups predates the creation of the laboratory, and the continued close collaboration with UiO has formalised an educational alliance between the two institutions; see the interview on page 499. In November 2005, Simula and UiO signed an agreement that explicitly stated the two parties would pursue the establishment of a dedicated school of research. The purpose of this school would be to educate outstanding researchers within the field of information and communication technology. In tandem with the discussion between Simula and UiO, the Norwegian government decided to establish funding for research schools[3].

During the planning of the school, it was decided to pursue both public and industrial support. In 2006, Simula and UiO approached a select group of industrial companies to explore whether interest existed amongst them for a collaboration in establishing a postgraduate program. The suggestion appealed to several companies, and a comprehensive proposal for governmental funding was sent to the Ministry of Education and Research in the fall of 2006. This proposal was based on an annual production of 15 PhD degrees. Given the framework of three-year projects, 45 PhD students should be active each year. In addition, the proposal defined a one-year position of a Research Trainee. This type of position is meant as a mechanism to identify whether a candidate is suited for PhD study, and whether Simula would be an attractive place for the student to pursue such an ambition. In order to feed the stream of new PhD students, 15 trainee positions per year would be required. Moreover, the proposal included ten postdoctoral fellows, five scientific supervisors, and three persons with administrative duties. Based on public rates for 2006, the annual cost of this enterprise would be 49 million NOK. At that time, Simula's annual budget for PhD and postdoctoral positions was 14 million NOK. To fill the financial gap, the proposal asked for an annual budget of 30 million NOK from the Norwegian government, provided that we could obtain five million NOK per year from industrial partners. In order to get started, the government was also asked to provide an initial funding amount of five million Norwegian kroner for 2007 to establish the school.

This ambitious proposal released the start-up funding of five million Norwegian kroner in December 2006, and SSRI was formally established in May 2007. Since its establishment, SSRI has received five million NOK annually from the government via the Research Council of Norway. This amount is, of course, far too modest to realise the ambitions of the original proposal. Still, in comparison, it should be noted that the government's total budget for Norwegian research schools in 2008, including Simula, was ten million NOK.

[3] In 2002, the concept of "Schools of Research" was introduced by an international evaluation of postgraduate education in Norway [2]. In 2005, the Parliament asked the Research Council to implement such schools [4], and the initiative was launched in 2008, subsequent to the start-up of SSRI.

In 2008, industrial partners supported the school with 5.2 million NOK, and the corresponding amount for 2009 is budgeted to be 9.5 million NOK. In addition, the Municipality of Bærum is contributing one million NOK per year for the four-year period 2007–2010. Independently of the establishment of SSRI, Simula has recently received several research grants from the Research Council, including the creation of a Centre of Excellence focusing on biomedical computing. These research opportunities include considerable funding for PhD students and postdoctoral projects.

Despite the still unfulfilled ambitions, the number of educational positions at Simula has increased significantly over the last few years. Compared with the 24 PhD students active in September 2004, 36 such students are affiliated with Simula in March 2009. In addition, there are currently 39 master students, one research trainee, and 16 postdoctoral fellows. SSRI is formally acknowledged as a research school at UiO, but it is not in a position to award academic degrees. Therefore, the degrees obtained by Simula's master and PhD students are awarded by a university, in most cases UiO, although collaboration with other national and international universities is also possible.

Industrial Partners Take an Active Role in Education. The industrial commitment in Simula's educational programme is clearly demonstrated through StatoilHydro's and Telenor's ownership in SSRI. These partners and the Municipality of Bærum contribute substantial funding dedicated to educational positions and scientific supervision; see the interviews on pages 495, 627, and 519. The long-term aspect of the financial commitment is particularly visible in the recently signed agreement with StatoilHydro that grants SSRI four million Norwegian kroner annually in the five-year period 2009–2013. This agreement is part of StatoilHydro's Academia programme, in which several Norwegian and international universities participate. It is complemented by other contracts with StatoilHydro that contain components of both basic research and technology development; see chapter 40 on page 553 and the interview on page 541.

More recently, Simula has signed an agreement with Det Norske Veritas that will dedicate ten million Norwegian kroner over the period 2009–2010 to research projects with a strong educational component. This initiative includes funding for one PhD student and two postdoctoral fellows, as well as internal funding for dedicated personnel on DNV's side. The topic for the collaboration will be software testing and verification; see chapter 28 on page 415 and the interview on page 405. Given positive results, the agreement can be extended for subsequent years.

Focus on operative and productive student-supervisor relations. As with the other scientific staff at Simula, the PhD students and postdoctoral fellows are expected to contribute research of internationally high quality. With respect to the scientific content, this aspect is the responsibility of the supervisor, in collaboration with the relevant project and department management. However, a critical factor in any supervised research that can affect both quality and progress rate is the relationship between the student and the supervisor. One particular goal of SSRI is to conduct elements of quality assurance regarding this relationship. Through a monitoring process, one aims at detecting potential problems early, such that proper guidance can be offered before a conflict or failure results. It is expected that in monitoring the students' progress via these actions, they will be better positioned to finish their degrees according to their original schedules.

The use of annual performance interviews and employment dialogues is central in the monitoring process. On the one hand, the performance interviews have a scientific profile, focusing on the achievements obtained by the student, and the strategy for continued work. These interviews are usually a formal meeting between the student and the principal supervisor, possibly involving the project or department manager. On the other hand, the employment dialogues are formal meetings between the student and a senior member of SSRI's administration. The purpose of this latter meeting is to chart the student's conception of his or her current situation, with respect to working conditions, supervision, and progress.

In addition to the individual meetings discussed above, SSRI also organises an annual seminar for the PhD students and postdoctoral fellows, and one for the supervisors. In such seminars, topics of common interest to these personnel groups are discussed. So far, our experience with the seminars indicates an increase in contact between students and postdoctoral fellows across departmental borders. SSRI also encourages and sponsors an informal meeting series with scientific content, arranged by the PhD students and the postdoctoral fellows. These meetings also offer arenas in which the groups in question can network.

Extended course portfolio teaches critical skills. The courses offered by universities focus on purely scientific skills. However, a successful research professional also needs additional skills to be competitive. In order to prepare our PhD students and postdoctoral fellows for the best possible careers, whether that be in academia, industry, or public service, SSRI maintains a portfolio of courses tailored to meet these needs. Currently, the courses address innovation and entrepreneurship, efficient communication of research results, and improved understanding of the research process.

In close collaboration with internationally leading experts at Pennsylvania State University in the United States, Simula has initiated the development of specifically tailored courses teaching scientific presentation and writing skills. These courses address needs at both the PhD and postdoctoral levels and are offered on an annual basis for 12 selected candidates, taught over two weeks by Associate Professor Michael Alley. In addition to the 12 students who benefit from this intensive practical training, the lectures in these courses are attended by all levels of scientific staff. In 2009, the first section of the presentation and writing courses was embedded in a national workshop on communicating scientific research; see [3, 5]. This workshop was arranged by SSRI and sponsored by StatoilHydro, Telenor, UiO, the Norwegian Defence Research Establishment, and IT Fornebu. Presentation and writing courses were offered to 68 participants from Norwegian universities, across all major fields of science and technology. At the deadline one month after the announcement of the workshop, there were more than 230 applications for these 68 seats. The feedback from the participants strongly supports the belief that there is a need for such national courses given at regular intervals.

Year	2001	2002	2003	2004	2005	2006	2007	2008	Total
PhD	4	4	2	5	5	7	6	4	37
MSc	15	13	21	19	24	24	24	27	167

Table 31.1 The number of PhD and master students who have graduated each year since Simula was established in 2001.

In addition to the courses on scientific presentation and writing, Michael Alley also teaches SSRI's annual course on efficient proposal writing. This course is aimed at both postdoctoral fellows and established researchers.

SSRI is also introducing an annual course on innovation and entrepreneurship. The first generation of it is being offered by Dr. Martin Edlund from Kebbison AB in Gothenburg, Sweden, and it includes exercises in which the participants will develop a business case.

As an example of activities targeting improved understanding of the research process, SSRI and the Basic Research unit have co-arranged the seminar titled "You and Your Research", based on a transcription of Richard Hamming's presentation that is reproduced in chapter 6 on page 37.

The supplementary course portfolio is continuously subject to revision and extension in order to adapt to the needs posed by Simula personnel. The next addition to it is expected to address research management and will be offered to postdoctoral fellows, as well as PhD students who are close to finishing their degrees.

What has Simula Achieved in Research Education?

We now have a steady stream of students finishing their degrees in connection with Simula's research projects. Table 31.1 shows the number of PhD students who have graduated each year since the laboratory was established in 2001. In total, 37 students have been awarded their doctoral degrees in the period 2001–2008, including 23 awarded degrees since the 2004 evaluation. In addition to the 36 currently active PhD students, several other doctoral students are scheduled to start their work later this year. In total, 10–12 students are expected to defend their dissertations in 2009.

By its very construct, SSRI shows how it is possible to actively involve leading industrial companies in the education of next-generation researchers. The organisation of the educational offer within a limited company co-owned by external stake holders is novel, and it has attracted considerable interest. Although Simula has not yet reached the level of public funding for PhD positions that will allow for a steady production of 15 new doctoral degrees per year, it remains a fact that SSRI is the largest research school in Norway.

The results of Simula's drive for research education are closely linked to the scientific success of the laboratory as a whole. Institutions conducting internationally acknowledged research of the highest calibre act as magnets, attracting the best students. As the international visibility of the research departments at Simula has increased, so, too, has the number of applications from highly qualified students increased. Today, Simula offers an international work environment that attracts strong, globally oriented candidates for positions at all levels. The international flavour of Simula is, to a high degree, a result of a diverse cohort of PhD students and postdoctoral fellows. Currently, the employees at Simula represent 18 nationalities.

The establishment of SSRI's course portfolio thus has been well received by PhD students, postdoctoral fellows, and supervisors. Already at this early stage, it is evident that this supplement to regular university courses adds value to the learning experience. By refining and extending the portfolio, Simula and SSRI will become even more attractive destinations for young researchers in the future.

What Comes Next for Simula's Research Education?

Educational efforts at Simula will continue to be focused on the development of SSRI with the vision of *providing excellent research education.* In the realisation of this vision, research scientists who graduate from SSRI will be expected to have excellent scientific skills and to perform research of high scientific quality. They will address important problems; they will have been trained in topics of relevance to industry and society; and they will have acquired excellent skills in professionally communicating their own research results. Further, throughout their training period at Simula, the scientists will have learned to acknowledge that a true scientist strives to live up to the highest possible ethical standards.

While a vision acts like a lighthouse in the dark that guides ships safely to harbour, there is also need for more concrete and measurable tools for navigation. SSRI has set its goal of offering an internationally respected, top-quality research education within the scientific fields inherited from Simula. When this educational programme is fully operative, the annual production of PhD candidates should be doubled, reaching a level of 15 completed degrees per year. SSRI will continue to foster independent and cooperative researchers with high scientific and ethical standards. Moreover, the school will ensure close contact with relevant partners in industry and academia.

Scientific quality and academic partnerships. Securing scientific quality is of vital importance to realising SSRI's goal of providing an excellent research education. Therefore, SSRI must offer the best learning conditions possible for its students, so they can master the craft of science at the highest level of excellence. This goal can only be achieved in intimate collaboration with the research projects in Simula's Basic Research unit. In these projects, the candidates collaborate in writing articles with highly skilled researchers and publish their results in the most reputable journals and conference proceedings. In this way, SSRI lays the foundation for achieving consistently high scientific quality.

SSRI promotes close collaboration with selected academic partners. In addition to our main partner, UiO, alliances with other strategically important academic institutions will be cultivated. Currently, there are formal collaborations in place whereby faculty from other universities are employed in part-time positions at Simula, and SSRI students are allowed to graduate from universities other than UiO. These universities that host research groups that are of strategic importance to Simula include the Norwegian University of Science and Technology, the University of California at San Diego, Texas A&M University, and the Georgia Institute of Technology.

In order to provide society and industry with highly skilled research professionals within information and communication technology, SSRI depends on access to excellent candidates. In close collaboration with the departments in the Basic Research unit, SSRI recruits highly qualified candidates internationally. In addition, at present, the majority of the employees at SSRI are male. It is important for SSRI to increase the number of female candidates, and it will do so by encouraging female students to pursue a research career within ICT. As a step in this direction, SSRI has already established a collaboration with a high school that runs a special programme for attracting female students to science.

Within this framework, both students and postdoctoral fellows at SSRI contribute to the main objective of SSRI: to educate research scientists at the highest international level possible.

Maintaining relevance to society and industry. SSRI actively supports Simula's mission statement: *"The research will focus on fundamental scientific problems with a large potential for important applications in society."*

By engaging in discussions and collaborations, and by cooperating with industry, SSRI addresses research topics of high relevance to industrial sectors and increases the probability of achieving an industrial impact. One way of interacting with industrial partners is to spend time in their environment and thereby obtain first-hand knowledge of the challenges they face. Such interactions are already well-established with the two major industrial owners of SSRI, namely, StatoilHydro and Telenor.

In addition, society at large faces many challenges that are relevant for Simula to address. In the same way as we engage in industrial challenges, we will engage in those that pertain to the public sector

Ensuring successful PhD projects on time. The position of research trainee serves as a means to carefully analyse personal skills and research topics. The goal is to identify an interesting research topic with a high potential for success, and to find a good match between appropriate personnel and research tasks. During the research trainee period, both the trainee and the research group will determine whether there is a common basis on which the trainee can pursue a doctoral degree. It is also a goal to have the trainee establish early links with industry or other research groups, typically through an extended placement in an external environment.

Simula supervisors are expected to meet certain requirements, with respect to providing professional feedback for the student's work. Supervisors will receive training in best practices for supervision of PhD students and should thereby be in a position to offer excellent conditions for research education. During the time the student works towards the PhD, a systematic evaluation of progression, both with respect to experimental scientific work and academic course work, will be performed. Students should be apprised of the expected parameters of excellent performance, and if deviation from the proposed plan for the work necessary occurs, an appropriate, revised course of action will be developed.

The environment offered by SSRI is student-friendly, and all students are included in Simula's company culture. Both socially and professionally, the students have access to the activities and the benefits included in Simula's working environment.

Strengthening postdoctoral education. A specific goal for SSRI is to contribute actively to both the scientific and personal development of personnel in educational positions. While students and faculty share a common understanding of the goals set for study towards the degree, many academic institutions tend to have a vague definition of the role played by postdoctoral fellows. However, SSRI intends to actively help its postdoctoral fellows in shaping solid careers in research, whether that be in academia or in industry. Depending on interests, choice of topics, and specific skills, postdoctoral fellows at SSRI can move confidently in any of these directions.

Postdoctoral fellows are specifically encouraged to gain experience as supervisors, either for master students or for PhD students. In the latter case, they will participate on a team of supervisors that includes at least one experienced senior researcher. Also, postdoctoral fellows should consider the possibility of becoming part of a fundraising team and thereby participate in the process of writing research proposals. In addition to these activities, SSRI arranges a seminar series with industrial relevance, and also offers courses that teach pertinent skills, such as research management and the writing of research proposals.

Concluding Remarks

The education of researchers is an important goal for Simula. It is also one of the services that Simula is expected to contribute to society. The creation of SSRI as a limited company co-owned with industrial partners has shown a new way of conducting research education in Norway. The results that Simula has obtained so far are positive, although there is still lack of funding needed to realise the defined ambitions.

In the short time span of SSRI's existence, large institutions in Norwegian industry have embraced the opportunity to help educate next-generation researchers. Although this industrial involvement is highly desired, for scientific reasons, it indirectly poses a potential threat. This threat becomes real if the access to industrial funds reduces the possibility of substantial, long-term, public funding of graduate-level education. After all, industry can be vulnerable to economic conjunctures, and even large companies can experience difficulties in sustaining the type of longevity that government can offer. By joining forces and putting in their combined strengths, both industry and society will benefit.

References

[1] Research Council of Norway. *Evaluation of the Simula Research Laboratory. Report of the Evaluation Committee Investigation*, 2004. ISBN 82-12-02017-7.

[2] Research Council of Norway. *Evaluering av norsk forskerutdanning*, 2002. ISBN 82-12-01684-6.

[3] M. M. Sundet and A. M. Bruaset. Workshop: Communicating scientific research. http://www.simula.no/education/ssri/communication-workshop, 2009.

[4] Vilje til forskning. Stortingsmelding 20 (2004–05), 2005.

[5] M. Alley, A.M. Bruaset, M. Marshall, M.M. Sundet, and S. Zappe. Development of a national workshop to teach Norwegian Ph.D. students in engineering and science how to communicate research. *Proceedings of ASEE Annual Conference and Exhibition 2009*. American Society of Engineering Education, 2009.

32

THINKING OUTSIDE THE BOX

An interview with Ingolf Søreide by Bjarne Røsjø

"We are involved in the Simula School of Research and Innovation AS (the Simula School) because this collaboration is of great benefit to us. It is in line with one of our top priority development goals: the need to produce good PhD students who can become the leading scientists or engineers of the future" says Ingolf Søreide.

Ingolf Søreide is head of the oil and gas company StatoilHydro's Research Centre in Trondheim. The partnership with Simula was entered into in 2004, when Simula Innovation carried out a pilot project. This was later developed into a large research project on oil exploration with what was then Norsk Hydro ASA. The project was basic research oriented and included Hydro's providing funding for several PhD students and postdoctoral fellows over a five-year period.

In 2007 Norsk Hydro merged its oil and gas division with Statoil ASA to form the oil and gas company StatoilHydro ASA, with close to 30 000 employees worldwide and a stock market valuation of about 400 billion Norwegian kroner (1 February 2009). One of the company's ambitions is to be a world leader in exploration and production technology.

Researchers who deliver results

"When the Simula School was formally founded in May 2007, Hydro became a shareholder along with Simula Research Laboratory, the Municipality of Bærum, Telenor, SINTEF and Norwegian Computing Center. Why?"

"It started off as a successful scientific collaboration. We realised that this was an interesting research community that delivered good results within the agreed timeframes. When Simula's management started thinking about a research school it did not take us long to realise that this fit in well with our desire to strengthen the re-

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_32,
© Springer-Verlag Berlin Heidelberg 2010

cruitment of researchers. Technology development is one area where we have a competitive advantage, and for many years we have worked on facilitating the entry of more talented mathematics and science students to the universities. What we are doing at Simula is the next stage of recruitment, namely ensuring that the most capable students go on to do researcher training after taking their master's degrees. The point is quite simply that StatoilHydro needs highly skilled researchers who can further develop exploration and production technology in the future. We are simply making a virtue of necessity," says Søreide.

StatoilHydro spends close to 100 million NOK every year on different forms of collaboration with the university sector, and maintains particularly close contacts with the Norwegian Academy of Science and Letters, the University of Bergen and the Norwegian University of Science and Technology in Trondheim. "We have also enjoyed successful cooperation with the University of Oslo, and our association with Simula provided a welcome opportunity to strengthen relations with the UiO community," states Søreide. Simula Research Laboratory and the Simula School cooperate closely with UiO, which, amongst other things, has formal responsibility for the PhD programmes.

StatoilHydro chose to go in as part owner of the Simula School because it would lead to closer cooperation. "Moreover we were very keen to play a part in ensuring that the school actually came to something. This is not an investment that will give a financial return, but the hope was instead to produce researchers who could generate research of a high international calibre. Research students at Simula have already achieved success in that their work has been published in respected scientific journals."

Models produce results

From the start the focus of the partnership between Simula and Scandinavia's largest energy company has been on developing new technologies and methods for oil exploration and production based on simulation techniques. "We use advanced data processing and visualisation technologies to integrate data and information into our models. When we look for oil or gas, we basically build a model on the computer of a large subsurface region, or reservoir as we call it. We then use the model to find out if there could be any petroleum in the reservoir, and where the petroleum is most likely to be found. It costs several hundred million Norwegian kroner to drill one exploration well, so a lot of money can be saved if the models can help us to be more accurate in our search," explains Søreide.

When StatoilHydro has found an oil or gas reservoir and has reached the production phase a detailed model of the reservoir is built to test how to extract as much as possible of the resources stored in it.

"First and foremost, we provide funding for the Simula School, but we are also able to give scientific advice if the PhD student or supervisor so wishes. It is perhaps equally important that the students have access to brand-new data, such as that generated through the latest seismic collection techniques. We believe that this will play a part in accelerating research and the implementation of new technology. If researchers, on the other hand, study old data obtained using out-dated methods,

a situation that is not unknown, it takes longer to achieve the best results. Those researchers who are reluctant to cooperate closely with the industrial sector should consider the fact that their scepticism could be instrumental in slowing down research development," Søreide comments.

Transparency in return

For Søreide it goes without saying that the researchers at Simula should retain their academic freedom. "What we require in return is that the research findings should as far as possible be openly accessible. Ideally the researchers publish their findings, so that both we and others can use them afterwards. We believe that better results are obtained when we allow the researchers the freedom to develop their ideas. This is a school of research and innovation and the point of it is to inspire creativity, new ideas and innovation. In short, we want the researchers to think 'outside the box'. It is said that the oil industry has been conservative as regards making use of new technologies and methods, and there is probably a lot of truth in that. But, the challenges we face when it comes to the exploration and production of petroleum resources are just getting bigger and bigger. So we have to be more aggressive in our efforts to develop new technologies, for example, by using this partnership with Simula to provide fresh impetus and lead us more quickly into new and unchartered territory."

Søreide is pleased that the Ministry of Education and Research and the Research Council of Norway been so positive towards the research school and hopes that they will continue their support. "We will do everything we can to continue the support we give to PhD students and long-term research at the universities in general and are particularly pleased with the collaboration we have established with Simula Research Laboratory and the Simula School. We have every intention of continuing this partnership," Søreide concludes.

33

A LITTLE COMPETITION AND A LOT OF COOPERATION

An interview with Arild Underdal by Bjarne Røsjø

"The University of Oslo and Simula can be seen both as competitors and partners, but cooperation between the two is far more important than competition. We will always have differences of opinion from time to time, but even a good marriage has to be able to cope with discussions about the way forward, for example," says Professor Arild Underdal of the University of Oslo.

Arild Underdal is Professor of International Politics at the Department of Political Science, UiO. UiO is Norway's largest and oldest university. It was founded in 1811, while Norway was still in union with Denmark.

As Rector of UiO from 2002 to 2005, Underdal played a part in shaping the partnership with Simula. Underdal currently serves as a member of the boards of Oslo University Hospital and the Norwegian School of Management (BI), is chair of the interim board appointed to prepare the merger of the Norwegian University of Life Sciences and the Norwegian School of Veterinary Science and is a member of the editorial advisory boards of several international academic journals.

Simula and the University of Oslo have a partnership agreement which covers, amongst other things, the education of PhD students in the field of informatics. Every now and then discussions arise about which of the institutions the students should belong to. In addition, several employees of the University of Oslo hold joint appointments at both institutions and sometimes a situation can arise in which the university, for example, has teaching needs that are difficult to reconcile completely with the research plans at the centre.

"But overall the cooperation has worked very well. The establishment of Simula has given an important boost to both teaching and research at UiO too. As the rector of UiO, I welcomed the establishment of Simula, and the fact that the Norwegian authorities, with the support of private companies, provided enough backing to

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_33,

allow Simula quite a free rein and enough resources to function effectively as an international research powerhouse. Moreover I am impressed with how Simula's own employees have risen to the occasion; it is not always easy to succeed even if the basic set-up looks promising," says Underdal.

"It would have been good if Simula had been located in Gaustadbekkdalen right next to UiO, instead of out at Fornebu. It would also have worked well had Simula been part of the UiO's organisational structure. But, as I have already said, overall this is a happy marriage," he adds.

Cooperation in the interests of the university

Professor Underdal is also pleased by the fact that Simula has been so creative in its efforts to establish cooperation with business and industry. "This cooperation will also benefit the university. We have high ambitions in this respect ourselves and Simula has played a role in enhancing the university's own approach to cooperation. This has been partly thanks to the dynamic head of the Department of Informatics, who was also the first managing director of Simula," Underdal points out.

"Fears that cooperation with business and industry will adversely affect academic freedom are exaggerated," says Underdal. "It is a potential problem and one that we will take seriously, but the same is true of any commissioned work, whether it is commissioned by the public or the private sector. Again here the overall picture is that Simula has found an effective and expedient approach. Simula has found cooperation partners that do not only contribute funding, but who themselves also have a significant level of competence within research and development (R&D). Companies such as the telecommunications company Telenor, the oil and gas company StatoilHydro, and the risk management company Det Norske Veritas understand that they must allow the researchers freedom to develop if they are to reap the full benefits of the centre. The result is something other than what we would normally call contract research, where the organisation commissioning the research can be too keen to manage the project and sometimes may even try to interfere with the conclusions. I saw a couple of cases of that when I was rector, for example when a pharmaceutical company wanted to commission some simple experiments and then misuse the research findings in its own marketing activities."

We cannot just wait for Ronaldo to come to us

"How could the cooperation between UiO and Simula be improved?"

"I can't find many faults with the cooperation as it is. In the early stages I was concerned that Simula should be more active in terms of researcher training, but now they have established a research school in cooperation with the university and a number of other stakeholders. An important question now is whether we could perhaps be more successful in attracting the best international expertise. If we take a look at what the other leading international research institutions are doing, we can see that they are recruiting not only individuals but also small or even slightly larger research groups. We could perhaps enhance our ability to attract the most talented people if we were more aggressive when it comes to looking for suitable candidates, planning for and facilitating appointments. I can envisage a situation in which Simula and UiO could find a way of working with business and industry to achieve this. As things stand today, we put out a job advertisement, and also put out feelers via the academic networks. Perhaps we could learn something from the football clubs: they don't just sit there waiting for Ronaldo or the other best players in the world to come to them, but instead go out and actively look for the most talented people and present them with offers."

"Is there a need for more institutions like Simula in the Norwegian R&D system?"

"There is nothing to stop us establishing more independent centres of this type, but my main point is that we have already significantly improved our ability in Norway to concentrate our resources and nurture the best research groups. Here I am thinking of the schemes set up by the Research Council of Norway, such as the Centres of Excellence scheme and the Centres for Research-based Innovation (CRI). The CoE scheme, the CRI scheme and Simula are not identical triplets, but they do share many of the same features, such as relatively long-term and generous funding arrangements by Norwegian standards, and a management with far greater decision-making power than was common in the past.

"The difference is that Simula was established from scratch, while the CoEs and CRIs have been set up at existing institutions. But we should not exaggerate the difference, because new centres also have to be set up at the universities, and Simula, for its part, has to recruit from existing academic groups. Besides, I think a certain level of organisational diversity is a good thing," Underdal concludes.

Arild Underdal

34

ARE YOU PLANNING TO TAKE A PHD?

Survival Guide for New Researchers[1]

Aslak Tveito

Most people tire of school very early and are eager to get out into the real world to earn a decent living doing a decent job, but there are exceptions. And the rarest of the rare take it all the way to PhD level. This little note is written for those rare cases. It is written to make things easier for you because when you start doing a PhD you will enter a strange world full of idiosyncrasies that everyone will assume you understand even though they are not at all obvious. Hopefully, this note will help to prepare you. It is written with the exclusive aim of making life as a PhD student liveable and with any luck even enjoyable. It is written for fresh PhD students.

There is no set formula for taking a PhD. But, when you read this, you will realise that a lot of what you may be wondering about others have wondered about before you. Perhaps, by reading this, you will manage to avoid some of the pitfalls my students and I have stumbled into.

Being a PhD student can be the most wonderful position you can have at a university. All your intellectual capacity can be focused on one major task–your thesis. You will be given the time and opportunity to make a thorough analysis of your hy-

Aslak Tveito
Simula Research Laboratory

[1] This note was written about ten years ago in Norwegian and was intended as background material for a seminar where PhD students met to discuss their progress. Since then, the note has been reused many times and, this is a slightly updated and abridged version of the original document. See chapter 31 for more information on research education at Simula Research Laboratory.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_34,

potheses without the bothersome deadlines and irritating interruptions that are likely to characterise the rest of your career. Your time as a PhD student can be wonderful. Enjoy it!

About PhDs and Craft Certificates

Perhaps you are wondering what a PhD really is? Completing a PhD is like completing a certification in research. Just as carpenters must prove that they are able to build a house in order to become master carpenters, researchers have to submit a thesis to gain a PhD and thus merit the label "researcher". Of course, it is possible to build something without obtaining a craft certificate and to carry out research without a PhD, but both qualifications define a standard for the work society has reason to expect from these groups.

The most important part of a PhD is the thesis. It is a presentation of the results of the research you have done during your doctoral studies. Without these results, there would be no thesis and without a thesis, there would be no PhD. In the past it was common for researchers to spend a very long time writing their doctoral theses, often completely on their own. That is no longer the case. Today's students are supervised and they follow a set plan of study; the thesis is part of that plan. In addition, the students are required to take ordinary university courses to expand their general knowledge base.

Although today's students must attend courses and have a supervisor to support them, there is no doubt that a PhD student's life revolves around the thesis. Theses can be written in many different ways. They can be written as one text–a monograph–in which one particular area is analysed at great depth. This type of thesis is quite similar to a master's thesis, but a PhD thesis must, of course, be much more detailed and must present a much more thorough examination of the subject. The other extreme is to write a series of scholarly articles on a topic and then put them together to make a thesis. The full range of alternatives between these two extremes is also allowed; you can write a few articles with a long introductory text that binds the articles together. Or you can write a lot of articles accompanied by a brief introduction to the field.

Each of these methods has proponents and arguments to support it. However, there is a general consensus that a PhD must contain *new findings* and demonstrate the candidate's ability to carry out independent research and present the results of that research.

Is a PhD Something for You?

For a long time in Norway, taking a PhD was something only very few people were able to do. This is now changing. In most technical subjects and in the natural sciences, 10 to 20 per cent of those who take a master's degree go on to take a PhD. This is probably to a large extent due to the fact that studying for a PhD now takes the form of an organised researcher training and because funding opportunities have improved considerably. Practically everyone who takes a PhD in Norway receives funding in the form of a grant from a university or the Research Council of Norway. This means that PhD students have a full salary during their studies. Some may think the salary is not that good, but compared with conditions for doctoral students in other countries, Norwegian research fellows are very well paid and enjoy good working conditions.

If you are considering whether to start studying for a PhD, you should above all think about the sort of work you would like to do for the rest of your life. If you would like to do research, for example:

- at a university,
- at a university college,
- at a research institute, or
- in the research department of a company,

you will almost certainly need to have a PhD. It is, of course, still possible to do research at companies and institutes without a PhD, but the trend is clearly moving towards a situation where a PhD is necessary for permanent positions. Besides, the most interesting projects are given to employees who have PhDs. At universities and university colleges, a PhD has long been a requirement for permanent employment in academic posts. It is also an advantage to have a PhD if you are interested in working with advanced development projects, and these days there are also some positions in public administration for which a doctorate is considered desirable but is not a formal requirement.

My advice is that if you want to do a PhD and you want to work in research, you should give it a try. It is a good sign if you enjoyed the research part of your master's thesis. First of all, though, you need to be genuinely interested in your subject and to enjoy working long hours. You should not start a doctorate because it is easier to stay where you are than move to an unfamiliar environment, or because you don't feel like getting an ordinary job.

Step by Step

How do you go about taking a PhD? What are the most important steps? Let us assume that you have completed a master's degree and want to continue in the same or a related field. Let us also assume that you start on a standard three-year programme.

The first thing you have to do is to secure funding. You can do that by:

1. applying for a fellowship at a university,
2. applying for a grant for a Research Council project,
   or
3. applying for a personal research grant from the Research Council or similar institution.

You can do this in consultation with a supervisor, or you can apply on your own and then find a supervisor afterwards. How difficult it is to get a grant will depend on your chosen field and how good your results were in your master's degree. My impression is that those who really want to will eventually manage to obtain funding.

It is not the most difficult part of doing a PhD. Once your funding is sorted out, it is not usually difficult to find a supervisor, unless you have very special requirements. In some disciplines, research fellows can largely define their own work. Naturally, you cannot count on finding a supervisor who is sufficiently interested in what you have chosen to study. So, in some cases, it may be a good idea to draw up your proposal in cooperation with a potential supervisor to avoid investing a lot of energy in something that will never come to fruition.

Once your funding is in place and you have found a supervisor, the two of you must draw up a project plan together. You need this plan to be accepted into a study programme. It must set out your theoretical syllabus, and you will need to make a schedule for the work on your thesis. Anyone who has gone through this process knows that the plan for your thesis will change a number of times along the way. All the same, there is good reason to put a lot of work into it. Firstly, the plan will help give you some ideas about a direction for your thesis. Secondly, it will clarify your supervisor's expectations of you, and thirdly, working on the plan naturally leads to conversations that throw light on the nature of the student-supervisor relationship. Make sure that you are happy with the plans being made, that you understand what they entail and that the tasks ahead of you seem interesting. I would not recommend that you start a project that does not appeal to you. Studying for a PhD is so demanding that you have to like your subject if you are to succeed.

The first year of your studies is arguably the most critical. If you can make good headway with your theoretical syllabus and with the thesis itself, you have every reason to look forward to the next two years. During the first year, it is important to immerse yourself in your chosen subject, to establish good routines for contact with your supervisor, and to start working on specific research-related issues. It may be a good idea to get most of the theory work out of the way so that you can focus on your thesis during your last two years.

In the second year your main focus will clearly be the research. By this stage you must have specific questions to work on and you should have a clear idea of what it will take to come up with a result that is worth publishing. At the end of the second year, you should be able to submit an abstract to a conference and deliver a paper. You have to submit an abstract to present a paper at a conference. An abstract is a very concise summary of the findings you are planning to present. The organisers will read through it and decide whether they feel you should be given an opportunity to deliver your paper.

The third year will focus on research and thesis writing. This is the case both if you are writing a monograph or a collection of articles. As you move into your final year as a research fellow, you will have a long consultation with your supervisor to take stock of where you stand, and to work out what is needed to get you through the writing process unscathed so that you are able to reach your goal on time. You should take the initiative to set up this meeting yourself and you should push to make sure that what is required is set out as clearly as possible. A year may seem like a long time, but it will go by very quickly as your fellowship period comes to an end. So, it is a good idea to plan to finish three months ahead of time, to give yourself some leeway as your deadline approaches. The final fine-honing of results and wording often takes time. It is extremely important at this stage that you continue to work enthusiastically. I have seen several students get into a panic as the deadline for submission approaches and they start to wonder whether their work is good enough. Obviously, you bear the ultimate responsibility for what you submit, so you will have to rely on your own judgment. On the other hand, it should mean something to you when your supervisor gives you the green light. He does not want a thumbs down from the panel either.

Presenting your Thesis

The presentation of your doctoral thesis consists of two parts:

1. a trial lecture, and
2. the defence of your thesis.

Both parts will be assessed by a panel consisting of three members: two opponents and an internal panellist. Your supervisor will not be a member of the panel, but he will often be involved in the administrative work of the panel and will therefore be part of the whole process. In the first part of the presentation, you deliver a lecture on a topic that you have been given 14 days to prepare. The topic will not be in your own specialist field. The purpose of this is for you to show that you have gained broad insight into your subject and that you are able to get to grips with new issues and provide an overview of problems and findings in a field with which you were not previously familiar. There can be little doubt that this type of examination is highly relevant to the work situations that many candidates will encounter later in their careers. Nonetheless, it is reasonable to say that no one fails this examination. I have never heard of anyone whose trial lecture was not approved, although naturally the quality can vary. I generally try to set relatively broad topics for the trial lectures, but this is entirely up to the panel.

It is in part two of your presentation that the fun really begins. This starts with the candidate or the first opponent giving an introduction to the thesis. Traditionally, this introduction was given by the candidate, but new regulations state that it is the opponent who should begin. However, it is a very difficult job for an opponent to do, so it is common to come to some sort of agreement where the work is divided between the opponent and the candidate, as appropriate. Personally, I think it is a good idea to let the opponent give a brief introduction to the field, then have the candidate introduce his own findings and put them into the context of his chosen field.

After this introduction, the first opponent should give his comments and put some questions to the candidate. The first opponent usually has 40–50 minutes to do this and then the second opponent has 20–30 minutes. They discuss what they are going to say beforehand to avoid overlapping.

The discussion between the opponents and the candidate is always very exciting. There are family, friends, colleagues and students present in the room. Everyone wants the candidate to do well and is delighted if the candidate scores some points on the opponents. Actually, it is rare for things to get heated during these discussions, but they are nonetheless exciting because there is so much at stake for the candidate and supervisor. The opponents are also rooting for the candidate; they know several years of hard work have gone into the thesis. However, that will by no means prevent them from jumping on everything from serious mistakes to trivial typing errors and inaccuracies.

When defending your thesis, it is important to remain calm no matter how badly things seem to be going. It is far too late to correct mistakes. So, if you realise that you have overlooked something or other, you should admit it straight away. You may, of course, launch a counter-attack, but you should be relatively certain of winning before doing so.

If your thesis is a collection of articles, one or more of them may have co-authors. If that is the case, you must be prepared to answer all questions related to the articles. You must not hide behind your co-authors, but take full responsibility for everything contained in the articles. It is important to be aware of this when writing an article so that, already at that stage, you become so intimately acquainted with all aspects of it that you can give a detailed defence of everything it contains. When you present your thesis, it gives a very bad impression if you constantly quote your supervisor as having said this or that, and appear to have accepted everything at face value. That will not give the impression that you are a researcher who is mature enough to earn a doctorate. In other words, stand up for what you have written, but admit mistakes when you realise you have made them.

You have one big advantage and that is that you are unquestionably the one who knows your thesis best. An opponent is invariably a busy person. Even though he prepares thoroughly, you are the only one who knows absolutely all the details. You must make the most of this.

When you have succeeded in overcoming the opponents and everyone congratulates you on your doctorate, you will experience a sense of euphoria on a par with what you feel on other important occasions in your life. Speaking from my own experience, I was so convinced that it was my lucky day, that I stopped at a kiosk and bought three lottery tickets. Nothing came of that, of course.

What is Really Meant by Research?

Research is discovering something that no one else has discovered before. That, in a nutshell, is what research is all about. So, to do research, you need to find out everything there is to know about what has been done in your field before. Some of this you will learn by taking advanced courses and seminars. But that will not nearly be enough. You will have to make a thorough search of the literature by reading advanced books and research literature; a good way to start is by typing in the main keywords of your project at Google Scholar[2] to find appropriate papers and books. When defending your thesis, you must be able look your opponent right in the eye and say that, to the best of your knowledge, your findings are completely new. If one of the opponents

[2] http://scholar.google.com.

can come up with a reference from a well-known journal which, already in the title, reveals that they have solved your problem, you are in deep trouble.

Research also involves publishing your findings so that others are aware that you have actually solved a particular problem, so that they can concentrate on other questions for investigation and can base any further work on your findings.

It can undoubtedly be stressful to think that the whole process revolves around your understanding something that no one else has understood before you. It is a bit like trying to set a world record in the 100 metres sprint. Everyone knows, of course, that it is becoming more and more difficult to shave even a fraction of a second off the world record time. While films from 50 years ago show athletes who obviously had a lot of room for improvement in terms of both equipment and technique, it is difficult to imagine that this can continue. Presumably there must be some sort of physical limit to how fast a person can run 100 metres. Similarly, you can wonder whether there is anything left to discover in certain disciplines. That very thought struck me in full force the first time I visited the library at the Department of Physics at the University of Oslo. The library houses a huge number of journals containing research results from over several hundred years. Is there really anything more left to discover? Unfortunately, my career in physics was limited to a single lecture in the first physics course. There we were given a list of all the laboratory exercises in the course and that was enough to push me towards more theoretical work for the rest of my life. But physics is probably like all other disciplines; the more you understand, the more you realise how much you don't understand. Every article you write will give you enough ideas and raise enough questions for you to write at least three or four new articles. In other words, don't be worried that there won't be anything left to do. There will always be more to do. You just need to get started.

What does it Mean to Publish?

Publishing your work means making your findings known to the rest of the world. The most common ways of doing this are to:

1. write an article in a journal,
2. write an article in a book,
3. write an article and put it on your own website or on a website used by researchers in your field,
4. present a paper at a conference,
5. make a "poster",
6. write a book.

It is very important for research that findings are made public as quickly and as effectively as possible. Moreover, it is important for individual researchers to get their results published. Researchers are assessed in the light of their publications when it comes to job applications, promotions, funding for research projects, etc.

In some research communities, writing an article in a respected international journal is considered more prestigious than alternatives 2–5. In other disciplines, on the other hand, there are very prestigious conferences at which it is extremely difficult to get a paper accepted. So it is not very easy to say in general terms what is most respected in the world of academia. In your own field, your supervisor will no doubt be able to tell you more about the most prestigious channels.

At conferences, new results can be announced by giving a paper or hanging up a poster. A poster is quite simply a large poster on which you set out your findings. The poster is hung up in an appropriate place at the conference venue. Books are often published in connection with conferences in which speakers at the conference are each invited to write a chapter. These books are called proceedings. Very few PhD students get to write a whole book on their results. However, the first PhD student I helped supervise did exactly that.

When you write an article and submit it to a journal, the editor of the journal forwards it to two or three experts ("referees") in the field. They read the article carefully and write a report to the editor. The editor decides, in the light of the reports, whether your contribution merits being printed in his highly respected journal. The very best journals set very high standards indeed and return most of the articles submitted to their authors with a comment that they are not suitable. But otherwise it is most common for the editor to send the article back to you asking you to make some corrections to it, in line with the experts' comments. If this happens, there is hope. You are given a second chance. It is important that you make the most of it and pay serious attention to what the experts have written. That does **not** mean that you should feel obligated to accept all their suggestions. You are completely free to say that you disagree and then argue your point of view to the editor. But, to get the article accepted, you should follow the recommendations made by the experts and the editor insofar as you feel that it is academically defensible. Since the editor is almost certainly a busy person, it is a good idea to write a detailed cover letter in which you list what has been done differently in the new version, the changes you have made, the suggestions you have chosen to ignore and why you have chosen to ignore them.

Are the Experts Always Right?

I think anyone who has published a number of articles has discovered that statements from experts are of varying quality. When working on my own doctorate, I had huge problems getting one of the articles published. I was very disappointed because, in my opinion, this article was clearly the most important in my thesis. I was certain that it would be accepted immediately. But that was not the case. One of the experts was very negative and had, in my opinion, completely misunderstood the whole thing. We solved the problem by sending it to a different journal where we had better luck. But we learned enough from this incident to realise that it is not sufficient to have good results, we must also be able to express them so that readers can manage to place the contributions in a larger context.

Your First Article

Usually, you write your first article with your supervisor. Your supervisor will endeavour to teach you how to write a scholarly article. Together, you will draw up a preliminary outline and discuss how much of what you have done you want to include in the article. It is often difficult to determine the degree of detail to be included, but your supervisor will have experience of this from his own earlier articles.

Whole books have been written on how to write articles. Your supervisor will no doubt refer you to some of them and will give you further advice based on his personal experience. The library can almost certainly also help you to find suitable literature on your topic.

There is, however, one aspect I would like to point out as being especially important. Articles are generally based on one or perhaps a few basic ideas. These must, of course, be developed and examined over many pages. Your results and reasoning are often far more complicated than the idea from which they were generated. *But even if your arguments get complicated and lengthy, you must make sure that you get your basic idea across.* It is the idea itself that is your most important contribution. It is the original idea that the reader must grasp first and foremost and that he then may understand how to apply to one of his own research problems. In order to explain the idea you should not be afraid to give quite mundane examples. That is completely fine as long as it helps the reader to understand. Naturally, your arguments must be watertight and not mundane, but far too many researchers do not put enough importance on helping their readers to understand the core ideas underlying their arguments.

Your Supervisor

What can you expect of the person who accepts responsibility for supervising you? This is a question that bothers many PhD students, especially when progress is slow and their grant is running out. Even though there are no set answers, some factors are clear. A supervisor should:

1. give you an introduction to the most important problems in the field in which you will be working,
2. give you ideas for topics for investigation to get you started in the field,
3. discuss the issues you will encounter as you work with the material,
4. teach you to write scholarly articles,
5. be aware of the most important conferences/ books/journals in the field in question,
6. know what it takes to write an article in this field,
7. introduce you to his contacts in the field,
8. read what you write with a critical eye - this also applies if you write a thesis that he does not co-author,
9. give you regular evaluations about how you are doing,
10. give you encouragement when you need it,
11. criticise you when you need it, and
12. ask you to drop out if he feels it is not going well enough.

The last point may sound brutal. No supervisor finds it easy to suggest that a PhD student find something else to do. But you cannot simply take a job as a research fellow, hold on to it for three years and expect your thesis to write itself. You will have to work seriously and purposefully all the time. If a supervisor sees that it is never going to work, i.e. that the candidate lacks both talent and the ability to complete a PhD, then he should say so.

Formally speaking, a supervisor cannot ask a PhD student to drop out. Research fellows have temporary employment contracts and it takes a great deal to break them. On the other hand, there is no doubt that a supervisor plays a key role in deciding whether the student's work qualifies for a PhD. If a supervisor goes to the extreme step of asking a student to drop out, it is because he is convinced that the progression is so poor that it is virtually impossible to believe that it will lead to a thesis. Obviously, such a request must by no means come as a surprise to the student. A supervisor must give repeated warnings over a long period of time before saying that it definitely does not seem to be working and that there is therefore no point in continuing.

The Student

As a PhD student, you must:

1. keep your supervisor informed on a regular basis of what is going on; it is a good idea to write an email every Friday to let him know what you have done during the past week. This will make things clear both for you and your supervisor and it shouldn't take more than about 10 minutes,
2. let him know if, for any particular reason, your work capacity will be reduced for a while
3. report your progress honestly–tell him what is going well and what is going badly–your supervisor should not only be given the good news,
4. throw yourself one hundred per cent into the discussion about the project–do not allow your supervisor to conclude every discussion–take part and demand to be heard,
5. say so if you think your supervisor's messages are not clear enough, he spends too little time on academic supervision, etc.
6. if you are not entirely satisfied, say so sooner rather than later, instead of waiting until you are really unhappy about something,
7. DO NOT expect miracles–take responsibility and say what is your mind; do so clearly so that you are sure that your supervisor gets your message,
8. work hard–do not fool yourself into believing that you can take a PhD and at the same time be involved in any number of other activities. You simply won't be able to manage it; your thesis should be the last thing you think about before you go to sleep in the evening and the first thing you think of when you wake up. You need to be totally immersed in it and this is something that will soon be very apparent to those close to you. They should understand that that is the way it is going to be,

9. find relevant articles, conferences and websites that contain information about your topic and forward this information to your supervisor,
10. help your supervisor; send pointers to literature, give summaries of important things you have read and so forth; it is in your best interest that the supervisor is updated,
11. acknowledge that your supervisor is busier than you are. So help to ensure that any time spent on your project is time well spent,
12. find another supervisor if you are not making progress and you think this is down to poor academic supervision.

Regarding the last point, I should mention, of course, that this is a dramatic step to take and that you ought to wait as long as possible before taking it. It need by no means be so dramatic in many cases. You can simply start working with another supervisor and thus have a gradual transition from one supervisor to another, without implementing a formal change. But let's be honest–some supervisor/student relationships flounder because of poor chemistry. If the problem is that bad, it is probably a good idea to switch supervisors–also formally.

Supervision Groups

So far, we have presented the entire process as a relationship between *one* student and one supervisor, but the picture is often more complicated. First of all, it is common to have more than one supervisor. This is especially desirable if the research fellow has a thesis that touches on several subject areas that need to be covered by two or more supervisors. Second, it is common for research fellows to cooperate closely and thus to a certain extent to supervise each other. It is obviously positive that a student gets supervision from several quarters, and establishing close working relationships with other research fellows can yield long-lasting benefits. But there are a couple of factors you need to watch out for. If you have several supervisors, you can be fairly certain that they will give you different messages and that it will be frustrating for you. This can be a problem, if they are from totally different research groups with completely different traditions for doctorates. But usually the best way to avoid such problems is to have joint meetings.

You should insist that you have a joint meeting at which you can set out examples of how you have received conflicting advice. You should not be afraid to create a bit of discussion or worry about things getting a little heated. It is essential for you that your various supervisors understand how you intend to proceed and why.

If you start cooperating closely with another research fellow, both of you should make your respective supervisors aware of this and you should tell them what you are contributing to the joint project. Your supervisor will need to know this to be able to judge whether you have contributed enough to defend your thesis.

The First Commandment

states "Thou shalt have no other gods before me." I do not intend to start a religious debate here, but want to draw attention to the sad fact that this commandment appears to be the guiding principle in certain research groups. These groups are usually characterised by a strong leader, whom the entire group follows almost blindly. In extreme cases, debate in such groups is characterised by the participants vying to articulate the “master’s” opinions. Once the master has expressed an opinion, the disciples will simply reiterate well-formulated versions of his statements. However, if the leader does not immediately express an opinion, the game instantly becomes far more complicated. The disciples’ job is then to guess, based on previous discussions, what the master is likely to think. Obviously, this calls for intelligence and some courage since the disciples may guess incorrectly. So, the debate in groups such as this usually only gets going once “the master” has spoken. At that point, all the subordinates can once again spout truisms.

Of course, such masters were once highly talented researchers themselves. Had that not been the case, they would never have attained their position in the first place. In the longer term, however, a culture based on ingratiation is the beginning of the end of a research group. A good brain needs to be challenged and a good researcher needs to be criticised. For that reason, new researchers do their supervisors a disservice when they kowtow to them. They do not need that. They need to be challenged. They do not need young researchers with second-hand opinions; they need new colleagues with new and challenging thought processes.

For that reason, you should never automatically accept a claim simply because it comes from your supervisor. Nothing is true simply because a particular person has said or written it. To the extent anything is true, it is true because it has been rigorously proven or because the arguments for it are so convincing that you cannot find any grounds for doubt. Who puts forward the argument is irrelevant. The standards that determine the integrity of a line of reasoning always apply. No one is so important or so famous that they can come out with poorly documented claims without expecting some debate. So you should, without exception, always be critical to everything you hear and read. You must, of course, also allow yourself to be convinced, but not until you have accepted the arguments in your own mind.

This can, of course, be taken too far. You can complain about everything under the sun, but that will not get you very far. Any supervisor will tire of listening and will end up giving you as little attention as possible. It is a good idea, especially early on in a doctoral programme, to accept your supervisor’s recommendations based on the supervisor’s experience and because you may be having difficulty keeping up. But you must not let this become a permanent state of affairs. You must eventually show yourself to be an independent researcher with well-founded ideas.

Full-Time or Part-Time?

There are many different ways to fund doctoral fellowships. Most doctoral students receive grants from the Research Council of Norway or from a university.

Research Council fellows usually have no other obligations than their studies, while university fellows usually have teaching responsibilities that account for 25 per cent of their time. Due to the teaching, university fellowships are for four years, while Research Council fellowships last for three years. Some Research Council fellowships are linked to a particular research institute so the research fellow is affiliated with that institute during his studies.

The research institute may want its research fellows to spend a certain percentage of their time working for the institute at the same time as studying. The work is generally related to research commissioned by the institute's customers. Universities like to have four-year research fellows with teaching responsibilities because this enables them to reduce the teaching loads of their other staff members. Similarly, research institutes use research fellows to reduce the workload of their researchers. It can clearly be an advantage for research fellows to get some teaching experience or experience with contract research. However, we often see that duties related to teaching or contract research demand too much attention and lead to a total derailment of the doctoral studies. There may be several reasons for this:

1. Pressure from the research group is greater for tasks for which the group bears a common responsibility. The tasks are specific and simply must be done. When things are busy, thesis work can readily be pushed aside.
2. If the thesis is progressing slowly, it is easy to focus more on teaching/contract research because the going may be a bit easier. It is only human to prefer doing what you feel you can do best, so the focus on your thesis may be less than it should be.
3. Expectations of research fellows' work capacity in terms of teaching and contract research are often overly high. There are often established norms for how much time any given project will take, but those norms usually apply to experienced researchers. An experienced lecturer may spend only 20 per cent of his work capacity on a given course, but a similar course may take up 50 per cent of the time of a new researcher.

My advice to you as a research fellow is therefore:

1. If you can, choose a three-year fellowship that focuses exclusively on doctoral studies.
2. If you have to take on other responsibilities, be very careful when signing an agreement. Talk to research fellows who done it before and listen to their advice. Stay away from research communities known for drowning research fellows in extra work. Unfortunately, such groups exist at universities and research institutes alike.
3. Make sure your supervisor understands the obligations you have, and get him to help you assess their extent.
4. Try to resist being monopolised by employers or students–stay focused on your thesis. You will be evaluated on the basis of your thesis and that is what you are there to write. Everything, absolutely everything else, should take lower priority.
5. Do not be tempted to take paid employment while you study. If you are going to accept paid work, it should be at the end of your studies when you are in full

control and certain that you are going to reach your goal. Be aware, though, that a lot of people have fallen into this trap. They think everything is done, but then some new problems crop up all the same. The best thing is to wait to take on paid work until after your thesis has actually been sent to the printer's.

The Research Council

As a new researcher, you will soon come across the Research Council. The Research Council may fund your project, or possibly you will come into contact with it when you apply for funds to attend a conference while you are studying. Regardless of what your first meeting with the Research Council is like, you will have heard a great deal about the organisation from more experienced researchers in the canteen or the break room. There is not a lot researchers agree on, but believing that research councils are populated by incompetent idiots who just want to overload you with tiresome administrative procedures and who are neither interested in research nor able to distinguish the good from the mediocre, seems to be a popularly held opinion among researchers the world over. It is actually quite interesting to hear researchers, who, as far as I know, have never led a single research project, slate the research councils' hopeless bureaucratic procedures.

I feel all of that is a vast oversimplification, so I would recommend that you, as a new researcher, form your own opinion of the Research Council. Approach its procedures, forms and guidelines with an open mind. They are not nearly as bad as they may appear at first glance. The Research Council has obviously put some effort into streamlining applications to make it easier to process them. This may mean that some of the questions seem slightly irrelevant to your own application, but don't worry about that. Just plough through the torrent of words, complete the form and send it in.

A large part of the bureaucracy the Research Council requires of its applicants and project managers is due to the fact that they themselves must report on how they spend money. They need good arguments to convince the funding authorities that it is important to support research. Also bear in mind that there are always a lot of applicants for funding–many more than there is funding available for, so you need them more than they need you.

But, there is one mitigating feature of the Research Council's work that I particularly appreciate. That is that they are interested in new ideas and they accept that those new ideas must come from researchers. I submitted my very first funding application when I was employed by the Centre for Industrial Research. Along with several others, I helped to draw up a highly ambitious project proposal. We asked the Research Council to fund a three-year programme expected to cost roughly 30 million NOK. That was a huge allocation at the time. After the funding was granted, a lot of people asked me who had suggested that we should apply for funding for our project. The question implied that people believed such a major initiative had to have come from inside the Research Council. That was and is wrong. It is not the job of the Research Council to come up with good ideas for research projects. It is your job or, more generally, the job of researchers. It is the Research Council's job to choose from the proposals they receive. To do this, they consult with well-qualified international experts, who make recommendations about which proposals should receive funding. So, my advice is that if you are going to apply for funding from the Research Council, you must, first and foremost, have a good idea of what you want to do. The Research Council funds good ideas. It is not enough to be a talented, well-known researcher, you must also be prepared to take the trouble to write down what you actually plan to do and why you want to do precisely that.

I don't believe that the Research Council can manage to spot all the good ideas; they have almost certainly rejected a lot of applications that, in retrospect, should have been given funding. All the same, I believe that, on the whole, they do a good job; just like the rest of us, they can only do their best.

35

AN EXTRAORDINARY INVESTMENT

An interview with Odd Reinsfelt by Bjarne Røsjø

"It is completely out of the ordinary for the municipality of Bærum to have invested in Simula, because it is not the type of activity a municipality usually invests in. The fact that we have done it all the same is because Simula fits in very well with our own business development aims," says Odd Reinsfelt, Mayor of Bærum municipality.

Odd Reinsfelt has been Mayor of Bærum since 1992 and, after his re-election in the local elections of 2007, is Norway's longest standing Conservative Party mayor. The party received some 40 per cent of the votes, an indication of the fact that the population of Bærum is well satisfied with both Reinsfelt and his party.

"We have no expectations that the investment in Simula will give us a direct financial return. Most of the municipality's investments, such as in schools and nursing homes, tend on the contrary to *incur* expenses. The municipality also owns shares in certain businesses, but then we are talking about businesses such as employment agencies for people who have fallen out of the normal job market. In other words our investment in Simula is something quite unique," explains Reinsfelt.

Skills and education

Bærum has invested one million Norwegian kroner in shares in the Simula School of Research and Innovation and Simula Research Laboratory. In addition, the municipality has agreed to contribute one million Norwegian kroner to the running of the business every year for a five-year period. The reason for this investment is that Simula fits in extremely well with the municipality's desire to develop the Fornebu area. "As far as education is concerned the municipality still has responsibility for primary and lower secondary school, but not for upper secondary school and in any event not for university level education. But our view is, nevertheless, that we want to facilitate the development of a forward-looking, high-tech and ideally internation-

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_35,

Odd Reinsfelt

ally oriented community for industry, research and education at Fornebu. Simula is exactly the type of player we want to have at Fornebu," says Reinsfelt.

Simula and the many other knowledge-based businesses that have established themselves at Fornebu—such as Telenor, Det Norske Veritas and Aker Solutions—offer a type of employment that is well suited to the population of Bærum. The population of Bærum (which is located on the coast, west of Oslo) is the most highly educated in the country; 43.3 per cent of the municipality's population over 16 years of age have a university or university college education[1].

Reinsfelt emphasises the fact that the municipality sees Fornebu and the Lysaker area, on the eastern side of Bærum bordering the capital Oslo, in the same context. Lysaker is both a traffic intersection and one of Norway's fastest growing business areas. It takes in part of the area known as "Engineering Valley," where high-technology companies are located along Norway's busiest stretch of the E18 motorway. The combination of international and urban Lysaker and the international but at the same time rural Fornebu area is seen as an engine for the development of the whole municipality.

"Bærum municipality has also been working towards establishing a university college or a university at Fornebu, but so far has not been completely successful in this. Simula, however, cooperates closely with the University of Oslo, and it was beneficial for us to be involved in its development and to establish a contact network," states Reinsfelt.

A very good relationship

According to Odd Reinsfelt, the relationship between Simula and Bærum municipality is very good. "We have noticed that representatives of Simula talk about the relationship with Bærum in positive terms, and that goes both ways. The fact that the municipality's contribution is after all relatively small has not been the cause of any upset. Rather, its involvement is seen as a show of goodwill," says Reinsfelt.

Bærum municipality was a strong proponent of the IT Fornebu concept right from the start. "The idea behind IT Fornebu was to create a synergy between high-tech companies, research and education, and in this it closely resembles the municipality's own goals. Moreover, Bærum is keen to support activities that add value, and I am not just talking in terms of financial value. Instead I am talking about human values, the type of values that enable people to be happy in the municipality and that make it exciting, right and important to develop themselves," Reinsfelt explains.

"You may have noticed that we have not put a local authority official or politician on the Simula board. Instead we asked one of the large companies in Bærum—the investment company Umoe—to provide someone to represent Bærum's ownership interest. In this way we were able to use our involvement in Simula to strengthen the good relationship the municipality has with the business sector too," he adds.

Reinsfelt thinks that, in general, countries other than Norway are better at establishing collaboration between local government and local businesses. "We have not seen much of the ongoing cooperation that led, for example, to such rapid develop-

[1] Statistics Norway, education statistics for 2003.

ment in Hamburg, where local government and business and industry collaborated on a huge push forward. In Norway parts of the business sector are unfortunately more concerned with furnishing their shareholders with money, an approach that does not foster development to the same extent. There are examples of successful partnerships in smaller municipalities where there are only one or two prominent business players, but in municipalities with a more diverse business sector such as Oslo, cooperation between local government and business and industry is more fragmented. Bærum is very keen to prove that closer cooperation is advantageous to both sides," Reinsfelt adds.

36

SIMULA CAN DO MUCH BETTER!

An interview with Morten Dæhlen by Bjarne Røsjø

"Simula's development since its establishment in 2001 has been amazing, but there is still plenty of scope for improvement! What I mean is that the most successful among us are those who recognise their weaknesses and work systematically to overcome them," says Professor Morten Dæhlen.

Professor Dæhlen knows Simula better than most, having been the centre's first managing director following its establishment on 1 January 2001. Dæhlen was later head of Simula Innovation from its start in 2002 to 2004, before he returned to the University of Oslo in 2005 and became Head of the Department of Informatics. So he was involved in developing both Simula itself and UiO's partnership with the newcomer.

"Simula has brought innovative thinking to the Norwegian research system, but in my view Simula's most important contribution is the fact that it led the way and showed how a new type of research centre with more targeted research priorities could work. It has gone very well and now we are seeing a number of university groups applying the same principles, in slightly different ways, to generate more research of a high international calibre," says Dæhlen.

As Dæhlen sees it the establishment of Simula was a realisation of prevailing ideas at the time. Many Norwegian researchers had talked about reviving the concept of the full-time researcher, researchers who could immerse themselves in their subject and who would not have to spend large parts of their working day applying for research funding. Many had also talked about focusing efforts on the most competent research groups instead of spreading limited resources thinly across the board. A funding system based exclusively on applications can quickly result in a situation where the commitment to focusing intensely on the best research groups

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_36,

is undermined: funding is allocated to weaker groups in the hope that these will become as competent as the best research groups.

Not particularly original

"There was actually nothing startlingly original about our establishing Simula. What was new was that we were given the opportunity to do what we had talked about for many years–and we did just that! Ten years ago it would have been very difficult to do this at a university in Norway. Now, however, the universities and the Research Council of Norway have achieved a lot. Many university groups have put into practice ideas similar to those that were realised at Simula and some groups have also learned from Simula's experience," states Dæhlen.

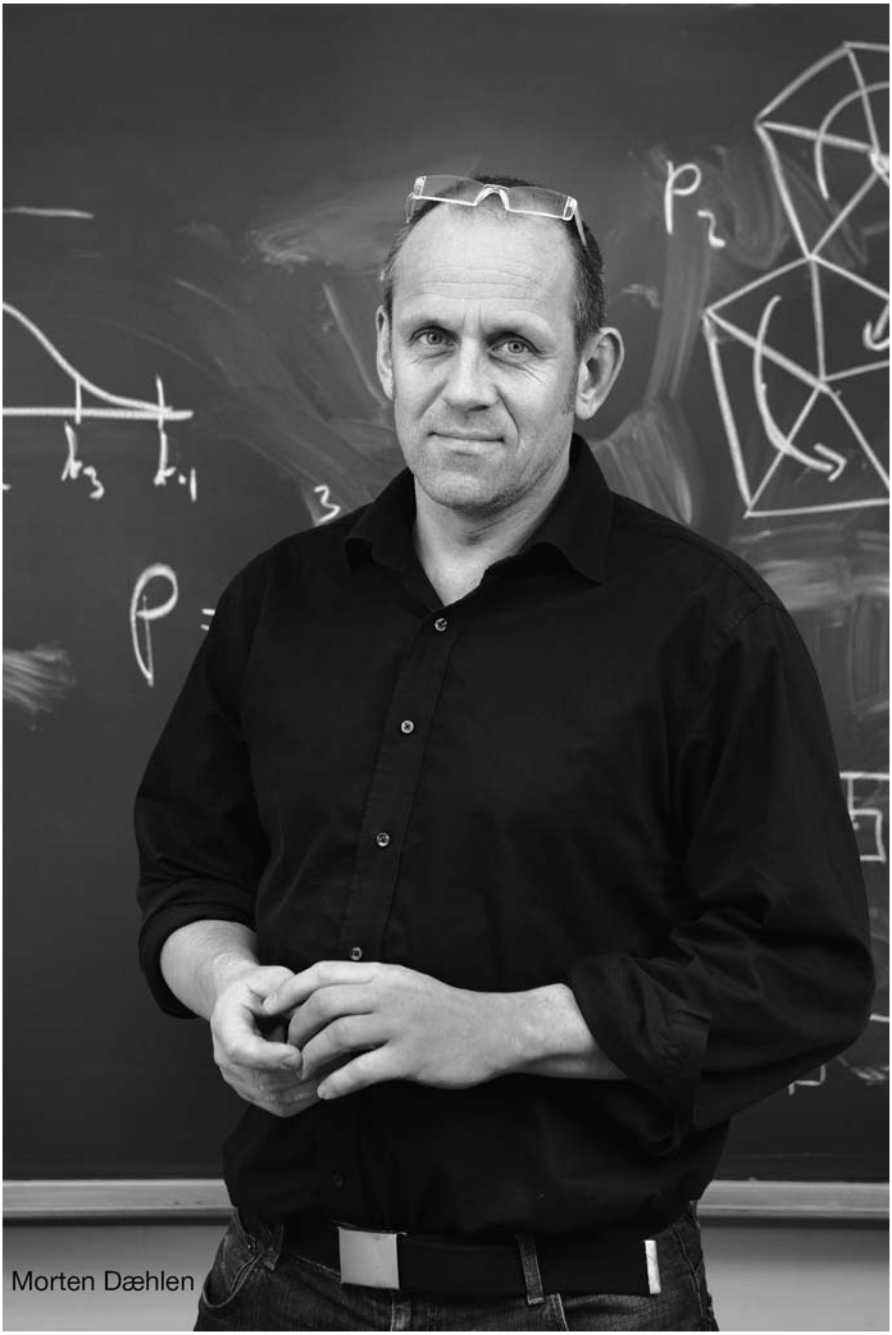
In 2001 the Research Council of Norway established a Centres of Excellence scheme, which was, to a large extent, based on the same thinking as that which lay behind the establishment of Simula. The CoEs were designed to concentrate resources and contribute to the development of more Norwegian research groups that could generate research of a high international calibre. "The CoE institutions share some common features with Simula, but Simula was in no way their source of inspiration. On the contrary, the CoE scheme was proposed in 1999 in a white paper on research presented by Jon Lilletun, then Minister of Education, Research and Church Affairs. At the time there were already a lot of centres of excellence in other countries," remembers Dæhlen.

Directed basic research

Professor Dæhlen is the type of person who has no problem juggling two or more thoughts in his head at a time. "Focusing on directed basic research, as we did at Simula, was completely the right thing to do, in that we started by identifying three areas in which we intended to excel. At the Department of Informatics at the University of Oslo it is not as easy for us to work in that way. We have a broad responsibility when it comes to teaching, which means that we cannot use all our resources on a few select areas. We also need to protect our academic freedom and ensure that we have enough resources to focus on new areas so that we are able to develop the discipline in line with the needs of society. In this respect we perform a task that Simula can benefit from in the slightly longer term, in that it can select and further develop aspects of the topics we have begun to cultivate at the university. The universities' institutes could also be more targeted in their efforts, but we always have to try and find a good balance between breadth and depth," emphasises Dæhlen.

"What could Simula do better?"

"Simula could be better at most things!" replies Dæhlen. "Perhaps the most important consideration is that Norwegian ICT policy and research policy in general is inadequately supported. Simula should become more involved in trying to change this policy. But this criticism of Simula could just as easily be directed at the Department



Morten Dæhlen

of Informatics and at me personally. We have not been good enough at maintaining the pressure when it comes to research policy either," Dæhlen responds.

One of the great weaknesses of Norwegian ICT research policy is that the level of funding has remained unchanged for years. "Even if we factor in the establishment of Simula, there have not been any increases to speak of. In the 1999 white paper, ICT was recognised as one of five focus areas. The other four areas have received some extremely substantial budget increases. By comparison, allocations to ICT have stood still since 1999," Dæhlen points out.

Professor Dæhlen has one more challenge for Simula: "Simula has chosen to focus on three subject areas and that was the right choice to make. If you are going to excel, you have to make a long-term commitment to focusing on your selected target areas. But, as I see it, we are losing out by not establishing mechanisms that enable us, every five years, for example, to explore new and great challenges. I think that the outside world would like to see us make new and important decisions from time to time. One of the most important responsibilities I have at the university is to ensure that the Department of Informatics reaches a position where it can do that, but it is a position Simula should also work towards. It is not difficult to identify new and important areas that need exploring. What is more difficult is to fund new research projects within already existing systems," Dæhlen concludes.

37

ACHIEVING RELEVANCE

An interview with Ine Marie Eriksen Søreide by Bjarne Røsjø

“I believe that cooperation between industry and the Norwegian research community should be, and will be, closer in the years to come. Simula provides a very good example of how this can be achieved,” says Member of the Norwgian Parliament, Ine Marie Eriksen Søreide.

“The mutually beneficial relationship between research and industry can partly be attributed to the fact that a large number of researchers are dependent on commercialising their ideas, which requires close cooperation with business and industry. The benefit for the business sector is that it acquires expertise and ideas for new products, services or methods. It is difficult to imagine how this collaboration could work without good contacts between the two sides,” Søreide points out.

Ine Marie Eriksen Søreide has taken a keen interest in education and research throughout her entire political career. In 2001, when she was elected Deputy Member of the Storting for the Conservative Party of Norway (Høyre) she went straight in as member of the Standing Committee on Education, Research and Church Affairs. After the 2005 general election she became a full-fledged member of parliament and was made chair of the same committee. Søreide is a qualified lawyer and was leader of the Young Conservatives of Norway from 2001 to 2004. Søreide takes every opportunity to stress that industry and research can be mutually beneficial to one another.

“In Norway, we have not come very far in this area, but I believe that collaboration between the research community and business and industry should, and will, be better in the years to come. Today collaboration is to a large extent dependent on the efforts of a number of enthusiastic individuals, whereas many other countries have come a lot further. The divide between industry and research is greater in Norway than in many other countries. This is due in part to the fact that research communities in other countries are far more dependent on external financial support, which leads naturally to closer cooperation with companies. Norwegian research is, to a greater

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_37,

extent, publicly funded. But Simula is a good example of the fact that cooperation between research and industry is possible, also in Norway, and is proof that this cooperation can yield very good results," says Eriksen Søreide.

Also seeking excellence in education

Høyre's draft election programme for 2009–2013 includes a commitment by the party to focus on developing excellent study opportunities by making use of first-rate scientific communities, using the Norwegian Centre of Excellence scheme as a model. The CoE scheme was established by the Research Council of Norway in 2001. It aims to improve the quality of Norwegian research by providing particularly competent research groups with long-term and generous funding. Høyre wants Norwegian researcher training programmes to be steered more in this direction too.

"In my opinion it is exciting to think that we could implement an initiative focusing on achieving the highest standards in education too. We have already seen a reform of the quality of Norwegian higher education, but the primary objective there was to ensure that more people completed their studies and graduated more quickly. The reform did nothing to nurture the most capable students or promote the best courses of study. I believe that an initiative focused on creating 'Centres for Excellence in Education' would have a profound and positive effect, also on when the talented students go on to become part of the research community," explains Eriksen Søreide.

Research Schools as an example

According to Eriksen Søreide, Simula has also led the way in this respect. "When, as Minister of Education and Research, my fellow party member Kristin Clemet presented the white paper 'Commitment to Research' in 2005, she launched the concept of research schools. At the time I remember a lot of people wondered what she meant by research schools. It was very good to be able to refer to Simula, which was already well on the way to defining what a school of research could be. Simula created its own model in close cooperation with industrial stakeholders. Now the Government Commission for Higher Education, known as the 'Stjernø-utvalget'[1], has proposed that all researcher training programmes in Norway should be organised along the lines of research schools and schools of research are gradually beginning to be established at a number of educational institutions. So a lot has happened in just a few years," Eriksen Søreide notes optimistically.

[1] The Commission for Higher Education (chaired by Professor Steinar Stjernø) was appointed by the Government in May 2006 and presented its report including recommendations for restructuring higher education in Norway in January 2008.

Ine Marie Eriksen Søreide

"There are some researchers who fear for the independence of research, if industry gains too much influence."

"Yes, a lot of people think that industry will be able to set the agenda for research if cooperation is closer. But this is not borne out by experience at all. Instead experience indicates that both academics, in general, and researchers, in particular, are good at safeguarding their integrity. One way of doing this is to include appropriate clauses in contracts for commissioned research. Researchers are generally good at making sure that they retain ownership of the research results, and that they are free to publish any findings, even those that are not favourable to the commissioning company or organisation. If the trend instead were such that closer cooperation made hostages out of researchers, I would agree that there was cause for alarm. All the same, there is every reason to keep an eye on this issue."

"Not all companies are bad American tobacco producers though, are they?"

"No, exactly! We should have basic confidence in the fact that the motives of business and industry are honourable. Nowadays, for example, there are a lot of companies that want to carry out research into how to be more environmentally-friendly, which is a goal we can all endorse."

Søreide is also concerned with the fact that the fight for the best brains is intensifying across Europe. "It is interesting to see that Simula has managed to attract a large number of research fellows from abroad. It has managed to build up an environment that is attractive even by international standards. Moreover, we are seeing similar developments at other places in Norway: a lot of institutions, particularly within the fields of science and ICT, now have more foreign PhD students than Norwegian doctoral candidates. In this respect, initiatives like research schools or Simula are very valuable because these are institutions based in Norway that at the same time have international appeal."

Some Norwegian research groups may have found it difficult that a newcomer like Simula has achieved such success over a period of just a few years, but Søreide has a different way of looking at it. "I think, on the contrary, that the University of Oslo, for example, which awards the doctorates for Simula and is a traditional Norwegian university, has benefited, because Simula has demonstrated that there are other ways of doing things. For my part I think that the way Simula is organised and its way of thinking represent the future of Norwegian research to a large extent. We will continue to focus on basic research, but within a large number of areas, the research needs to be more relevant and applicable than it is today. Organising researcher training along the lines of research schools and the Simula model is the most effective way of achieving this," Eriksen Søreide concludes.

PART IV

RESEARCH APPLICATIONS

Audun Fosselie Hansen, Director of Simula Innovation, and Ottar Hovind, Director of Administration and Research Applications.

38

BRIDGING THE GAP BETWEEN INDUSTRY AND RESEARCH

Are Magnus Bruaset and Marianne M. Sundet

Why Should Simula Care About Applications?

For Norway to stay at the economic summit and continue to prosper, our knowledge-driven industry must grow and improve its capability of competing on the international scene. To reach such goals, there is urgent need for research that can provide solutions to the technological challenges faced by individuals, industry, and society. Being a laboratory at the international frontier of research in information technology and communication, it is a natural consequence that Simula "...will focus on fundamental scientific problems with a large potential for important applications in society." In tandem with conducting basic research and educating researchers, Simula's mission is to promote the application of the research in both private and public sectors. The importance of industrial cooperation, innovation, and business creation is clearly visible in the corporate structure that Simula implemented in January 2008. The *Research Applications* unit accounts for all these types of activity and houses also the subsidiary Simula Innovation AS, which constitutes the laboratory's main instrument for innovation and entrepreneurship.

"One problem in Norway has been that the gap between the research and educational institutions on the one hand, and business and industry on the other has been to large," states Paul Chaffey, the Managing Director of Abelia, the business Association of Norwegian knowledge- and technology based companies, in an interview

Are Magnus Bruaset · Marianne M. Sundet
Simula Research Laboratory

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_38,

on page 89. Chaffey explains that so far this problem has been solved by having a large number of independent research institutes operating somewhere between research and industry, whose focus has largely been on applied research. To create market dynamics in applied research, this has been a good system. But at the same time, this system has also given rise to a problem in that pure research groups at the universities have lacked exposure to industry and the market. According to Chaffey, Simula provides a solution to this problem by showing how the gap between pure research and industry can be bridged. "Simula has produced research results that could not so easily have been generated at an applied research institute. This is because the institutes' dependence from an industrial sector that can pay for the findings makes it difficult for them to implement extensive and long-term research projects," he explains.

Simula has a great point of departure for the task of bridging the gap between industry and research. The Research Applications unit is co-located with internationally leading research groups, and benefits from a tight scientific integration with these groups. Due to the high scientific standing of Simula as a whole, the laboratory stands out as an attractive partner for innovation and commercialisation activities. Successful combination of research and innovation has resulted in the implementation of large, long-term industrial collaborations with major companies, as well as the establishment of several commercial spin-off companies. In relatively short time, Simula has achieved a high level of awareness for innovation and business creation among the researchers in the laboratory, and candidates for new spin-off companies are continuously revealed and examined.

How can Research make it to Business?

Since Simula's establishment in 2001, innovation has been regarded as a cornerstone of the organisation, along with research and education. In course of Simula's development, the understanding and interpretation of the term "innovation" has matured. In contrast to traditional academic environments, where this term often is regarded as a business slogan invading the researcher's work, Simula has developed an ever-present awareness of the duality between basic research and its application to both industry and society. This process has positively influenced the majority of the senior researchers, and their change of mindset is continuously shaping the attitudes of the research staff recruited more recently.

In June 2004, SI was established as a separate limited company, fully owned by Simula. The main motivation of SI is to increase the return of national investment in Simula by promoting and facilitating the application of Simula's research results. To this end, SI assists the research departments at Simula in their efforts to realise their results in terms of applications. Thereby, SI contributes to Simula's strategic ambition of generating at least one major commercial breakthrough by 2015. The access to in-house innovation expertise with hands-on technological experience is of great advantage in meeting this goal.

At Simula, the outlets of innovation have taken on different forms: large, long-term industrial collaborations; pilot projects investigating the commercial potential of selected research results; and the establishment of commercial spin-off companies, possibly co-owned by external investors. In particular, industrial collaborations with StatoilHydro and Telenor have evolved into long-term, research-based alliances that offer highly visible and attainable possibilities that can have a significant impact on science and business. Both collaborations described herein go beyond joint publication of academic papers. The output from these collaborations has either an industrial impact through its use by our industrial partners or it possesses significant potential for commercial spin-off.

The coupling between academic research tasks and technology development has proven to be bidirectional, in that development-oriented activities lead to new PhD and postdoctoral research projects, and vice versa. The intimate connection between the axes of research and development is a particular strength in the StatoilHydro-Simula collaboration. This feature is a consequence of both companies being committed to long-term research, and the flexibility offered by Simula's organisation. In total, the funding has been divided more or less equally between research and development activities. Inside Simula, these types of activities are split between Simula School of Research and Innovation and the commercial subsidiary Kalkulo AS. The supervision of PhD students is conducted by senior researchers in the Basic Research unit. It should be noted that all types of StatoilHydro-directed activities are under the same project management. StatoilHydro and Simula view the collaboration to be of strategic importance, and both companies plan for long-term continuation and extension of the activities.

Complementing the industrial collaboration, a total of seven spin-off companies have been established, and a number of commercialisation projects have been conducted. Among these achievements, Kalkulo AS stands out as a successful spin-off company that is deeply involved in the collaboration with StatoilHydro. The more recently established company Lividi AS shows a strong indication of establishing itself as an important actor in the market for scalable video streaming.

Although the innovation activities in total have been successful, there is still considerable room for improvement. The experiences that have been obtained form a solid basis for further design and revision of Simula's mechanisms for technology transfer. An evaluation of the most successful and promising projects and spin-off companies, thus far, has shown that they share a mix of the following characteristics:

- The innovative idea has root in the core activity of a basic research project in one of the research departments at Simula.
- The research activity has a strong foundation in a strategic collaboration with a major industry or financial partner.
- The commercialisation process has been a close collaboration with other technology transfer offices (TTOs) like Birkeland Innovation at UiO, incubators like IT Fornebu Incubator, or other commercialisation experts.
- The researchers are the key assets in the commercialisation projects and spin-off companies, due to the fact that the technologies, products, and services are extremely knowledge-based.

What are the Commercial Achievements of Simula?

Large, long-term industrial collaboration. In 2004, a promising contact was made with Hydro's oil and gas division[1]. This contact has since emerged as a long-term research collaboration in Computational Geosciences, firmly based on the expertise and results accumulated in the Scientific Computing department at Simula. By the end of 2008, this collaboration has accounted for a total funding of 32 million NOK. The collaboration continues in 2009 with increased strength and has a defined scope until at least 2014. The research collaboration with StatoilHydro stands out as a hallmark in terms of funding and activity level, covering the full range of possibilities from education and basic research activities to commercial development of state-of-the-art technology. Further details about the collaboration with StatoilHydro can be found in chapter 40 on page 553 and in the interviews on pages 495 and 541.

The main goal for the collaboration with StatoilHydro is to strengthen the work procedures in oil and gas exploration through new and improved computer-based models describing geological and geophysical processes. Thus far, two novel, research-based technologies have been successfully established through this collaboration. These results are referred to as the 4D Lithosphere Model and the Compound Modelling technology. Currently, a patent application covering the core algorithms in the lithosphere model has been submitted, and the compound modelling software is in use for both research and production cases in StatoilHydro. Even more contributions are in rapid progress. These contributions will likely lead to improved reliability of depositional models, better insight into the physics of underwater gravity flows of sediments, and more accurate descriptions of deformations in sedimentary basins. All these applications concern important steps in the workflow employed when evaluating a geological region in order to assess its potential for production of oil and gas.

In 2007, SI initiated a strategic research and innovation collaboration with Telenor Research and Innovation[2] within the fields of telecommmunications, information technology, and media. The resultant SimTel project provides an exclusive opportunity to perform innovative research within the fruitful partnership of an academic research institution and a large network operator with its strategic vendor partners. While the research aspects are taken care of by the Networks and Distributed Systems department, the commercial aspects are handled as a project run by SI. Education of the PhD students in the project is organised through SSRI with supervisors from the ND department.

The goals of the SimTel project are to establish joint research activities, and establish and follow up on joint applications to the Research Council of Norway and the European Commission. In addition to publishing joint papers, information exchange between Simula and Telenor researchers will be established and supported. The project aims to continuously evaluate the commercial potential in the joint re-

[1] After the merger between Hydro and Statoil in October 2007, the original partnership with Hydro was continued in the new company StatoilHydro. According to Forbes Magazine, StatoilHydro is currently the 53rd largest company in the world.

[2] Telenor Research and Innovation is the innovation hub for the Telenor Group. Telenor is the world's seventh largest mobile provider with services in 13 countries across Europe and Asia.

search activities, and identify related commercialisation potentials in both Simula and Telenor, as well as to nurture ideas and innovations from Simula in which Telenor could be a pilot or sponsor. The SimTel project will also apply for commercialisation funding from the Research Council of Norway and Innovation Norway.

Examples of other industrial collaborations. In addition to the large industrial collaborations with StatoilHydro and Telenor, Simula is involved in several other projects together with industrial companies. To exemplify, we herein consider the collaborative efforts undertaken together with Sun Microsystems, Inc. (Sun) and Det Norske Veritas.

Sun develops and provides a diversity of software and microelectronics that power everything from consumer electronics to developer tools and the world's most powerful data centres. Sun's European design centre for Netra Systems and Networking is located in Oslo and has a close relationship with the ND department at Simula. Over the years, this relationship has resulted in several joint papers and the inclusion of a Simula-developed routing algorithm in Sun products. Based on the positive experience derived from this relationship, Simula and Sun signed an agreement for the joint research project Sunrise in October 2008. The Sunrise project will conduct fundamental research in interconnection networks for high performance computing and represents an excellent opportunity to combine academic research with industrial relevance. It has an initial duration of three years and it will fund one postdoctoral fellow and two PhD students. The total budget is 7 million NOK, whereof 3.5 million NOK is provided by Sun.

During 2008, there have been several discussions between Simula and DNV concerning the possibility of establishing a research collaboration. DNV is a leading international provider of services for risk management, and conducts research in several areas, including biorisk management, future energy solutions, information processes and technology, maritime transport, multifunctional materials and surfaces, IT risk management, and software products. Recognising that information technology is an integral part of modern business, DNV has a strong interest in methods and tools for managing the risk of complex IT systems. The company is also a major vendor of advanced simulation software, typically used for stress analysis of large-scale structures. Therefore, there are natural connections to the research conducted at Simula. Thus far, the primary interest is in a collaboration involving the Software Engineering department at Simula, in particular one that will address methods and tools for software testing and verification. In April 2009, the two companies signed an agreement stating that DNV will fund one PhD student and two postdocs for two years. Given a successful outcome, the agreement is meant to be continued. In addition to the funds provided for educational positions, DNV agrees to allocate substantial internal resources for the collaboration. The total value of the agreement for 2009–2010 is ten million NOK.

Spin-off companies. Simula is involved in the creation and build-up of several companies that are based on research conducted in the laboratory. Simula's policy concerning spin-off companies is organic, in the sense that it is natural to set the company free if or when its business grows in a direction that falls outside the laboratory's fields of interest and competence. On the other hand, Simula takes an active part in the development of the companies for which its expertise is seen as useful. In total, seven spin-off companies have been established, with Kalkulo AS and Lividi AS standing out as particularly successful cases. For further information on these companies, see chapter 42 on page 613 and the interview on page 601.

Kalkulo AS was established in 2006, offering consulting and tailored software development in scientific computing. This company has a special strength in the fields of geometrical modelling and advanced visualisation. The senior personnel represent probably the most experienced and versatile competence of this type in Norway and holds an internationally high standard of acclaim. This type of competence has a wide applicability in the field of scientific computing, and there are close connections between the company and the SC department at Simula. The company has had a healthy profit each year since it was launched and is regarded as a successful spin-off from Simula. Currently, the company has nine full-time employees. Of these, eight are working as software developers. Simula owns 100 per cent of the shares in Kalkulo.

Lividi AS was established in 2008 and offers video streaming solutions for content providers and network operators. Lividi is a company that has spun off from the ND department at Simula and the Department of Informatics at the University of Oslo. The company is owned by the founders, together with SI and Birkeland Innovation. The Lividi team consists of eight persons. Currently, five persons work for the company on a daily basis, of which three are full-time employees. Simula owns approximately 36 per cent of the shares.

What is the Next Step Towards Bridging the Gap?

The researchers at Simula continue to produce a high number of innovative ideas with potential for commercialisation. However, SI has limited financial and organisational power to take innovation projects into the more demanding commercial phases. Therefore, SI's strategy is to establish close links with venture capital firms, industrial investors, and alliance partners. SI is also establishing close connections with larger technology-transfer offices and incubators in the Oslo area. For example, some of the most promising projects and spin-off companies are currently supported in close collaboration with Birkeland Innovation and IT Fornebu Incubator. Many researchers at Simula also hold a position at UiO, and there is a common ground for many of the research areas addressed by the two institutions. Therefore, SI collaborates closely with Birkeland Innovation on such common projects. Since Simula is located at Fornebu, it is also natural for SI to collaborate closely with the IT Fornebu incubator, which opens up more specific incubator funding opportunities. SI is also discussing governmental-supported alternatives for seed capital with other TTOs in the Oslo area.

It is one of SI's goals to further strengthen the focus on commercialisation of the key results derived from the basic research conducted at Simula. SI will contribute to a stronger focus on applicability of the research in each of the three research departments. Strategic industrial partners such as StatoilHydro and Telenor are extremely valuable in order to define exciting projects and exploit the results, as both companies also hold strong competence on innovation and commercialisation. Inspired by the existing links to industry, SI will assist in establishing more strategic industry partners that are relevant for the research projects in the three departments. This initiative is based on the theory that industry involvement in the early stage of research will foster solid commercialisation projects with a high potential for success.

The volume of high-quality projects generated by Simula over the past several years has been very high. When SI was established in 2004, the generation of sufficient ideas with market potential was seen as a major challenge, especially with Simula's modest size taken into consideration. Since SI was established, the close collaboration with the researchers at Simula has been essential, and SI's main focus has been to assist the research departments at Simula in their efforts to realise their insights in terms of real-world applications. As a part of this interaction, SI has focused on building a strong competence on the phase of idea generation and precubator services through the EFFECT model and the T2M model. However, even with a certain volume of projects with market potential, the road to a commercial success is still a long one, and success is by no means assured in the current market. One key element will be the right management resources. Given its limited resources, SI needs to concentrate on a narrow set of commercially promising projects. For the future, this implies that SI might not take a very active ownership in all projects and companies, but instead might try to find other partners, such as industrial collaborators, TTOs, and incubators that are willing to take on that type of active role. In these instances, SI will still follow and influence the development of the projects and companies, for example, through board memberships.

Although SI today has a growing network, an important aim is to target contacts that, in addition to financial strength, also possess the pertinent expertise required for the successful outcome of commercialisation projects. One of the most obvious places to seek collaboration is amongst major industrial companies with a relevant portfolio. In addition, it will be very important to establish close connections to relevant TTOs and incubators. A list of venture capitalists should also be a part of SI's strategic network. However, the experience so far shows that it is challenging to attract venture capital for commercialisation projects in an early stage, in which the final product and business plans are still to be developed.

A key to success for knowledge-based spin-off companies will always be the commitment from the researchers. Therefore, SI will contribute to the process of making pure basic researchers more business-aware, and increase the competence in building up start-up companies. To stimulate researchers to join a start-up company, SI will evaluate different models that allow the researchers to rejoin the research departments in cases where the start-up fails or becomes mature and independent of the founders. In summary, SI has formulated several key strategic tasks on the basis of the accumulated experience. SI will:

- Continue to contribute to the researchers' increased focus on the applicability of their research.
- Concentrate resources on a few but promising projects with strong links to the core activity in the research departments.

- Secure appropriate resources to develop and guide the commercialisation projects.
- Increase collaboration with relevant industry, TTOs, and incubators.
- Assist the research departments in establishing strategic industrial partnerships and sponsored research projects. A future goal for SI will also be to establish knowledge-based spin-off companies, such as Kalkulo, for the other research areas at Simula.
- Support the researchers who consider a commercial rather than an academic career.

39

MAKING THE INVISIBLE VISIBLE

An interview with Are Magnus Bruaset and Bjørn Rasmussen by Dana Mackenzie

As a top manager for Norway's largest oil company, Bjørn Rasmussen has a chance to make decisions that can affect the lives of many people. Case in point: In 1991, Rasmussen was the regional head of exploration in Norway for Hydro. One of the geophysicists in his team, Terje Enoksen, found seismic evidence for a gas field off the coast of Norway, in a region where experts had not expected any more large finds. As Rasmussen says, the conventional wisdom at the time was that "all the big elephants are gone."

In strictest secrecy, Rasmussen arranged for a more careful seismic survey–so secret that only he and five other people in the company knew about it. At first the results were disappointing–in the new survey, the gas field seemed to have disappeared. But Rasmussen's team re-analyzed the data, and this time the field showed up again. Another seismic survey confirmed it.

By 1997, when the first test well was drilled at the field, now known as Ormen Lange, Rasmussen had moved on to become Hydro's head of research. Ten years later, in October 2007, the giant Ormen Lange gas field finally began production, with Norway's King Harald V himself attending the grand opening. Rasmussen estimates that Ormen Lange will continue to produce natural gas for more than forty years. In addition to providing energy security, the field has also given a huge economic boost to Norway. For example, the company had to erect the world's 16th largest hotel simply to house all the construction workers who were building the onshore processing plant. It has been an amazing economic return for one courageous decision, which only a few people even knew about, to keep on looking when common sense said to stop.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_39,

October 2007 was a time of momentous change in Rasmussen's life for other reasons, too. That month, Hydro merged with Statoil, the company where Rasmussen had started his career, to form StatoilHydro, western Europe's fourth-largest oil company. "It was very exciting to meet my old colleagues at Statoil again and show them that I had not retired!" Rasmussen jokes. Far from it! In the new company he accepted a position as country president in Angola. He now lives in Luanda, the capital of Angola. Once again he has the opportunity to affect the economic development of a country—and he has the responsibility that comes with it.

You might expect an oil executive to be mostly interested in economics. But nothing seems to make Rasmussen happier than talking about geology. During our interview at Simula, he got up twice to draw diagrams on a whiteboard and show us some of the basics of petroleum geology. He describes himself as a "creative, entrepreneurial person." In order to keep his creative juices going, he loves to surround himself with technically skilled people with a different mindset. That is one of the reasons he decided in 2004, as senior vice president for global exploration at Hydro, to start collaborating with Simula on a variety of computational problems in geology and geophysics. The collaboration has continued to grow after the Statoil merger, although Rasmussen himself is no longer directly involved with the project. Another of Rasmussen's creative ideas was a series of "professor meetings," in a different country every year, in order to keep abreast of new ideas coming out of academia as well as from other oil companies.

Are Magnus Bruaset, who leads the Simula team that is collaborating with StatoilHydro, also participated in the interview with Rasmussen. We also conducted a separate interview with Bruaset, and some excerpts from that interview have been included in this one.

"Could you describe how the professor meetings got started?"

Bjørn Rasmussen (BR): "It started in 1994, when I took over the position as head of research in Hydro. I was not a typical researcher; I came from the exploration and operational side of the company, and the company wanted to focus on more customer oriented research. I immediately discovered that our company was not the only source of good ideas. We really needed to know what is going on out there, including at other oil companies. What are they doing at Shell and BP? They are not stupid at all. There are a lot of smart people out there. You just have to realize that and have respect for it. So I started talking with a lot of different professors in different environments.

"At these professor meetings, we bring together colleagues whom we are not necessarily acquainted with, from different disciplines within the geosciences. One person will know this and one person will know that, and so on. It's very good for creativity."

Are Magnus Bruaset (AMB): "I think these meetings are really a type of vitamin shot, both when it comes to the science but also for getting to know these people. They have strong personalities and of course they are very knowledgeable people."

BR: "Another important thing is that we meet outside the office, because when you're sitting in the office you have your jacket on, and you're going to be more formal… We take them out to relevant and nice places. Sometimes there is an opportunity to have a field trip at the end, and look at the rocks. We went once to Morocco. Last time we had the meeting in Luanda, and before that we met in Italy, and several other places."

"Where are you going to meet this year?"

BR: "Since I'm in a different position now, no longer head of research, somebody else needs to take over. Luanda was a transition. Of course I have a lot of suggestions for where we could go. I proposed Bilbao for this year. In Bilbao you have these very nice salt domes."

"Could you explain what those are? I have heard about salt domes underground, but I can't imagine what they would look like above ground."

BR: "Bilbao is quite humid, so the salt has been dissolved, but you can see there are remnants. The pillars start out underground, and over time they might be exposed at the surface. Of course as I said, there is a lot of bad weather there, so the salt will be attacked immediately, leaving just the rims around them at the surface."

"Is the salt imbedded in other rocks?"

BR: "It's mainly solid salts. It's unbelievable. The Atlantic Ocean, in the process of opening up due to plate tectonics, was at a certain stage a huge closed sea. So a lot of salt was deposited at the bottom–up to 1000, or in some places 2000 meters of salt. Can you believe it? All this salt, or evaporites, as we call it. Then more layers of sediment were deposited on top of it.

"If you tilt the sediment and the underlying layer of salt slightly, the salt will start to move downdip and upwards. You also have to remember that the density of salt is less than the sediments above. So the salt wants to get up. That's when it forms these salt pillars. You have them in Germany, and also in the southern part of the North Sea, on the British side. It's a really interesting geophysical phenomenon."

"How are they associated with oil?"

BR: "You have the salt and you have the layers stacked above through time. The salt we are talking about in Angola is 130 million years old, give or take a few million years. The salt starts moving, until it hits a cap rock, like this glass. (*He turns a glass upside down to demonstrate.*) This can make a good trap for oil."

"So the same thing that traps the salt will also trap the oil?"

BR: "The hat collects whatever is available of oil from the source rock. To create an oil deposit you need three things: source rock that the oil comes from, a reservoir where it collects, and a trap on top."

AMB: "At the same time these salt layers pose difficulties for detecting the oil and for drilling it."

BR: "Of course what I gave you was a simplified explanation. (*He walks to the whiteboard and sketches a typical salt pillar, topped by the 'hat' of impermeable rock.*) There's an opportunity for an oil field up there, on top of the salt pillar, under this hat. Also in the Gulf of Mexico as well as Angola, you have a lot of oil on the flanks. This salt is just pressing itself through the layers of rock. (*He draws the layers of rock that the salt has broken through in its buoyant rise toward the surface; at each point where it breaks through a layer, pockets are formed where oil can collect.*) But then the salt is sometimes really nasty, going out to the side like this. (*He draws a salt dome that is shaped like a mushroom.*) If you have oil beneath the salt dome, how can you find that? On top of the dome, it's straightforward with today's seismic techniques. But underneath, how can you see it? The seismic wave fronts go off in all different directions when they hit the salt. How can you get these wave fronts organized in a proper way? That's why you need massive computers to analyze the data. No one else is using so much computer capacity and processing and imaging today as the geophysical industry.

"We want to know: How can we *predict* these features? We established an internal project called 'How to make the invisible visible.' For several years, we even named the professor meeting after this, and we called it 'the invisible professor group'!"

"Let's talk about your collaboration with Simula. How did that come about?"

BR: "It was in Easter of 2004 that I met Morten Dæhlen. Actually I know the exact date, April 6, because I just checked it out on my calendar before I came here."

AMB: "I think it was through a friend of Morten's, Kjetil Solbrække, who at the time was the Senior Vice President for international development at Hydro."

BR: "We just spent a day off in Morten's cabin; I went down there and we sat with a few beers. It took some time to clean the brain. It's very dangerous to jump on an idea too quickly and then you just end up with nothing. But with this enormous computer power and Simula's knowledge of how to use computing resources efficiently, even though Simula had been working in different areas we thought it might be possible to do 4-dimensional backward geological modeling. Also, it was just nice to have somebody in Norway to build on. Simula had a world-class environment for this type of research."

Are Magnus Bruaset and Bjørn Rasmussen

"Why did you believe that Simula would be able to contribute to StatoilHydro, especially given that Simula didn't have a lot of experience in geophysical work before?"

BR: "We started up a pilot project, and we all got convinced that we had an opportunity here. Of course it's research, and there is some risk, but if it is really research there *should* be some risk. Otherwise it's not research, it's just development. There should be some failing. When I was head of the research centre, I used to say, 'If you're successful more than 50 per cent of the time, you're doing something wrong. Then you are not doing groundbreaking research.'

"Also, I thought, why not take advantage of this computer environment at Simula? I think Simula had 80 or 90 people at the time, so they had the capacity."

AMB: "Also we had the advantage that we direct our research. If we decide that we are going to do a thing like this, we set up a team for it. They will concentrate on that. That's a difference. Also, one thing I imagine has been a vital thing in this collaboration and different from more academic collaborations is that Simula can take care of both the basic research, such as PhD projects, and also do some more product-oriented development. The duality here is special."

BR: "Yes, that was very attractive. That's a good remark... In the beginning, I had to convince my own people. It's one thing for management to have a good idea, but you really need to discuss it in a proper way internally and make adjustments to the idea so that all hearts are beating for it. If not, you can forget about it. This takes some time. It was very good that we did the pre-project, so that key stakeholders in the exploration department got interested."

"What was the pilot project?"

BR: "It was about general understanding of modeling, geological processes and how to recreate today's landscape. After a while, the discussion broadened to include plate tectonic reconstruction, bringing us to 4-dimensional backward geological modeling. We discovered just before I left that you can use this as a database. We are always struggling with proper databases for maps. It was like an extra product that came out of it; which no one was really expecting."

AMB: "We had a really steep learning curve, understanding what this was all about. One thing that came out of the pilot study was that it gave Hydro the opportunity to go through the exploration workflows again. It became clear also where we wanted to put in the effort. For example, one of the things that came up over and over again in the pilot study was the need to do depositional modeling, simulating how the sand and mud particles settle on the sea floor. This is a type of modeling that has been asked for in the industry for years. There are tools that do this, but it's very hard to calibrate them directly, so people don't believe the results very strongly. It's a bit on the edge."

BR: "If you emphasize the reservoir issue, then how can you predict the reservoirs and their quality and location? You need to understand how a river system looked 100 or 150 million years ago. Sedimentologists have traditionally been very descriptive. They describe what they have: this is a beach deposit or these are turbidites or whatever. But how do you describe the reason why it is precisely there and not in some other place? My philosophy in exploration is that you want to know that information."

"Do you agree with Are about this phenomenon that people don't really believe the computer-based depositional models? If so, how do you overcome that and convince people that it will be useful?"

BR: "You need first to look for the enthusiasts and doers in your organization. Then you let them test it themselves. You cannot force people to believe in it. If you say as a manager, 'Use this,' then you can guarantee it will not be used. A good way to implement new software, particularly in the geosciences, is by jealousy. You get new software, and your next door neighbor looks at it and says, 'What's going on there?' If you give it to everybody, and tell everyone that they need training and courses and the whole works, they won't use it properly. If you give it to the ones who are really dedicated, then the others will say, 'Maybe I should have this, you know.' They'll come scratching at my door, maybe not today but in half a year. It sounds a little stupid, but that's the way it works."

AMB: "I think you're right. Seeing your neighbor get results is always very motivating in any field, not only in geosciences."

"Are, what was interesting about the pilot study from Simula's point of view?"

AMB: "Going back to the question about the use of computing techniques in geology, I think there is a division between the more production-oriented aspects and exploration. As seen from the outside, the production side has been dominated by mathematics and computers and an engineering approach for many years–for example, reservoir modeling, which mathematically speaking is typically porous media flow. On the exploration side it seems as if there are a lot of uncharted territories. That's one thing that made this possibility of collaboration exciting for us. There was a possibility to make contributions that would be important."

BR: "Exploration is a really expensive activity. It's really only for the big guys. This year in StatoilHydro we are participating in drilling about 65 wells, and each well costs anywhere from 30 million to 100 million dollars. Each one! We share some of the cost with other companies, but still it's a lot of investment. The question is: How can we be even more clever in predicting where we can find oil, and how much oil is there? We want to improve the rate of discovery–by doing the imaging in a better way, by understanding the development of the geology in a better way, by modeling.

If we can just increase the discovery rate by a small per centage, then by comparison the money we spend on a computer model is just peanuts."

"Do you see any progress in this direction? Simula is working on a 4-dimensional lithosphere model. Has StatoilHydro started using it already?"

BR: "The lithosphere model is a help, but there is still a way to go. I would also say that compound modeling has been very important. In global exploration in 2006 and 2007, we tried to use compound modeling on most prospects, just to have a quality check."

AMB: "The compound modeling technology was originally developed by Steen Petersen at the Hydro research centre in Bergen. To simplify it, you can describe it as a systematic way of taking measurements from wells or taking parts of seismic data that you really think are correctly interpreted, and propagating that information consistently through the total domain by using geological models. Originally he did this in a two-dimensional version, so you only had vertical sections of the geological layers. We started discussing this with him in 2004 and started working together in 2005. We were able to introduce numerical methods that made it possible to do this in three dimensions and do it very efficiently, so it's just a matter of seconds to compute these properties.

"The starting point for compound modeling is that you can take these sparse measurements and distribute values from those consistently through the domain if you're able to compute the distance between any point in the domain and the dominant geological objects, typically a surface (for example, a fold or a fault) or the borehole itself.

"Also in compound modeling, you look at the composition or the definition of a geological model as a sequence of geologic events over time. You can think of it as a software program for the creation of geology. By executing the series of events that is that program, you create the geology. Steen explained these things in our first session with him, in 2004, before we started the pilot project. I'm pretty confident that there was no one else in Hydro that understood the enormous potential of this methodology. We did not understand the full scope of it, but the ideas wowed me and my colleagues right away. So it has been very satisfying to work with Steen on developing this methodology."

BR: "Steen is a very interesting person. He's so strong on the technical side, and stubborn in a good way. I admire him a lot, actually."

AMB: "That was a very interesting experience from the pilot study. When we first heard Steen talk, I think we actually grasped points that other people missed. He thinks like a computer scientist, even though he's a geophysicist."

BR: "That's very much correct. If he explains something to me three times, then I'm with him!"

AMB: "When we met Steen, compound modeling had already been used for some prospects, but it has accelerated since then. The last time we talked about it, it has been used in 80 to 90 cases. For example, it has been used in Troll (Norway's largest oil field) and in prospects in the Middle East.

"What Simula has done with compound modeling is very much due to one person in my team, Øyvind Hjelle. One thing that came out over the first year or so was the use of the fast marching algorithm to compute distance fields. It was actually a small stroke of genius from Øyvind's side to see that connection and introduce that method. Since then we have put much work into refining that methodology. The literature describes it, but there are lots of caveats in the use of it that are not described, especially for such complicated geometries as we are talking about here. So he has really brought that methodology forward, mathematically speaking.

"Then we integrated this into the existing software for compound modeling, which has been refined over the years. At one point we had a situation where Øyvind was developing and implementing new features and continuously feeding them to the research centre at Bergen. Typically with a delay of six to eight weeks, they would already be sitting on the engineers' desks, being used for real cases! It was like having a pipeline straight into that analysis. You can really get scared by it, but on the other hand it's really motivating to have such an evident and immediate impact."

"Do some of the scientists at Statoil come to Simula and talk about what they would like?"

AMB: "Not that many, but there are a few we have collaborated really closely with since the beginning of the activity. Jakob Skogseid comes here very frequently. He's stationed at Vækerø, very close to Simula. He was the first project leader from the Hydro side to work with us. One of my fears in the beginning was that we would be talking in foreign languages without connecting. But from the start, when we began working in 2005, the communication has been excellent, not only with Jakob but with all the project leaders. That has been a necessary condition for getting things done properly.

"In early 2005, Jakob brought to the table this idea of the 4-dimensional lithosphere model. It was, as far as I understood, an idea that he had for many years and sort of put in his drawer in the office because he did not have an environment where he could realize the idea. But when he saw the possibilities here for doing advanced geometry modeling and visualization, then he took the idea out of the drawer again. Today, it is the technology that has come closest to becoming a shrink-wrapped product. It is being used by other research institutions. StatoilHydro has a collaboration involving Caltech, University of Sydney, University of Missouri, the Norwegian Geological Survey (NGU), and Simula. Each of these institutions will have PhD students or postdocs working on scientific problems using the lithosphere model together with GPlates and SPlates, two different software tools for manipulating the tectonic rotation models. Of course, when you get new people using these things, they will come up with their own ideas of what features should be in there, what should be done, how the user interface should behave. So we expect to get some good input from them.

"We had the kickoff meeting in August last year, and in February we had our second meeting. That was the first time the other institutions actually got full access to the software... the first out-of-the-house release."

"Can you describe the lithosphere model?"

AMB: "We start with a 3-dimensional visualization of data from today, but then combine it with a plate tectonic rotation model. So you can take these different grid-based data sets, maybe a whole stack of different surfaces and move them in space as you move the time slider according to an underlying rotation model. We are not developing the plate rotation model ourselves; that is something for tectonic specialists. But being able to move these data sets sufficiently according to the model, as far as we know, has not been done before. We use quite advanced visualization techniques combined with very efficient use of graphics cards, for instance. Also it is streaming technology, very much like Google Earth. In fact, that is a comparison I like to make. Take Google Earth and throw out all the tourist information. Add in all the geological features you would like to have, and instead of just looking at Earth's surface, let it go under the surface with a stack of layers. And finally, do this efficiently. Then you are in the neighborhood of what the lithosphere model is. Then add the fourth dimension of time, of course.

"The technology behind the lithosphere model has been realized by Christian Tarrou and Trond Vidar Stensby of Kalkulo. Before they joined Kalkulo, they used to work for a company that develops visualization software for weather forecasts on television. One of the users of this software is CNN. So they have world-class expertise in modeling and presenting time-varying data on the globe."

"Is this something that could eventually be sold as software?"

AMB: "There will definitely be possibilities to look into that. In both cases, the lithosphere model and the compound modeling, the idea of joint commercialization pops up now and then. For the compound modeling project, one of the deliverables for the project in late 2010 is an assessment for possible commercialization. StatoilHydro is looking into that, although no decisions have been made at this time.

"I see one use of the lithosphere model as educational. By education, I don't necessarily mean PhD students at a university, but education of professionals in the oil company. Very few people have an understanding of the interaction of the different scales, from the large scale of plate tectonics to the smaller scale of the basin or domain you're interested in. Those who do understand have a problem both doing the analysis and also communicating their insight, because there has not been a good way of visualizing these interactions. The lithosphere model can fill that gap."

"Bjørn, I understand that you are not an active participant in the Simula collaboration any more, because of your work in Angola. What were the reasons for your decision to move to Angola?"

BR: "When the two companies merged in 2007, I decided that I wanted to do something completely different. Angola is the biggest international unit in StatoilHydro, with two hundred thousand barrels a day in our pocket. It's a huge activity. We're participating in eight producing units. Besides these opportunities, we also have all these issues to deal with in a third world country, everything from poverty to corruption."

"How does StatoilHydro, when they go into a country like Angola, ensure that the benefits go to all the people in the country, rather than just to the government? Do you have ways of spreading the wealth around, so to speak?"

BR: "I think the Norwegians are generally very well regarded in the Third World, because we have a social democratic attitude in Norway. I think StatoilHydro is very much appreciated in Angola, also due to the fact that we have a solid social program supporting education (including university degrees), health and water sanitation, and environmental and technical projects (including geoscience and oil field development). Local and national content is also important to us. We collaborate with the authorities and the state oil company, for mutual benefit and to help them learn. Just as the Americans came to Norway in the 1960s to educate the Norwegians, we are doing the same now in Angola. We are currently the technical assistant for Sonangol (Angola's national oil company) in blocks 4 and 34 on field development and exploration.

"There are extremely clever people down there, but due to 27 years of civil war, they have lost two generations of education. The main issue, the main challenge in Angola is the lack of education. Twenty-seven years of war, you know... The young people were taken out of school, given guns, and then sent to war."

"Has the war ended?"

BR: "There was a cease-fire in 2002. The entire thinking of the bureaucracy and administration was focused toward the war, so it takes some time to change. It was even more than 27 years of war, because before 1975 there was the independence war against the Portuguese. I really admire the Angolans and give them credit for what they're doing. They're really moving ahead.

"So there are a lot of opportunities to participate in the development of a country, and to bring to the table some of the Norwegian traditions. A lot of delegations from Angola have visited Norway to learn how we do things in the oil business, and how to educate the local people in the proper way."

"Is the younger generation in Angola going to Norway or to Europe to study?"

BR: "Yes. Through our programs we are helping them out, sending them to Trondheim and to Aberdeen and so on. I just came from a meeting today, and we want to send more of them to Rio de Janeiro because they speak the same language. Sometimes you end up with a language barrier, but they are both Portuguese speaking countries, the same language on both sides of the Atlantic."

"Do you feel the same sense of responsibility in Angola as you did in Norway, the same sense that the things you do can affect the whole country?"

BR: "Yes, I hope so. This is part of why I feel good about working there. In the best case we might make a difference, because we are fairly well regarded down there as a creative, entrepreneurial company. We want to do the best for the country at the same time as we are making good money. As I mentioned before, we are the technical assistant for Sonangol in two blocks. In Angola there is a strong drive towards higher local or national content in all awarded contracts, so that the country can build up a local capacity. This is similar to how the situation was in Norway in the early days. The requirement for local/national content can create frustration in the industry. However, we from Norway understand this need and want to participate in a constructive and positive way. It is demanding, but those are the rules, you just obey them and try to make the best out of it. This is much more fun than fighting them!"

"Are there similarities between the Angolan deposits and the Norwegian? Has your experience in the North Sea enabled you to go to other places in the world and find things other people can't?"

BR: "Not necessarily. But Ormen Lange is deposited more or less the same way as one of the discoveries in Angola as well as in Brazil. It's kind of a lookalike. The main part of the reservoir in all of them consists of turbidites."

AMB: "When you look at the 4D lithosphere model, and you rotate things back to where they were 150 million years ago, you see that Angola and Brazil are two sides of the same deposit."

"What are your plans for the next few years?"

BR: "I really want to spend the remaining time of my career, if the company will allow me, in Angola. I'm getting up to speed on the language, and I like it a lot. You need to like the people, otherwise you won't be efficient down there."

"After you retire, will you stay in Angola?"

BR: "No, actually I bought a house in Brazil, up to the north, on the beach. We had one of our professor meetings there. That's where I plan to go when I retire."

40

TURNING ROCKS INTO KNOWLEDGE

Experiences and Results from an Industrial Collaboration in Computational Geosciences

Are Magnus Bruaset

Abstract Since early 2005, Simula Research Laboratory and StatoilHydro have built a strong and long-term research collaboration in computational geosciences. The main goal for this collaboration is to strengthen the procedures used in oil and gas exploration through new and improved computer-based models of geological and geophysical processes. So far, the 4D Lithosphere Model and a new generation of the Compound Modelling technology have been successfully established, and the collaboration has become strategically important for both organisations. More contributions to the field are progressing rapidly and potentially will lead to improved reliability of depositional models, better insight into the physics of underwater gravity flows of sediments, and more accurate descriptions of deformations in sedimentary basins.

This chapter describes the background for the research collaboration as well as ongoing scientific activities. Using the results obtained from academic and industrial pursuits as a backdrop, we also summarise the key factors responsible for the successful implementation of a close link between the basic research community and

Are Magnus Bruaset
CBC, Simula Research Laboratory

Are Magnus Bruaset
Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_40,

industry. The coupling of research tasks of an academic nature with technology development has proven to be bidirectional, in that development-oriented activities lead to new PhD and postdoctoral research projects and vice versa. The intimate connection between the axes of research and development is a particular strength in the StatoilHydro-Simula collaboration. This feature is a consequence of both companies being committed to long-term research and the flexibility offered by Simula's organisation.

40.1 An Industry in Change

When investigating the geological evolution of a given region, discussion between geologists often spurs new, competing theories. The rise of such concurrent theories draws attention to a characteristic of geology: A crucial point in this discipline is the *qualitative* understanding of physical processes spanning tens and hundreds of million years. Through extensive use of collected data that are subject to a high degree of uncertainty, individual experience and subjective interpretation of the targeted problem become vital components of this understanding.

The geological assessments routinely conducted in the oil and gas industry form the basis of many important decisions that can have potentially large impacts on the economy, environment, and society. Oil companies combine theories and tools from all facets of geoscience research. They also rely on disciplines with a stronger connection to mathematical models and numerical simulation (e.g., geophysics and geodynamics) than what is commonly found in geology. However, computer-based simulations and experiments play a larger role in the production phase, where oil or gas reservoirs have already been located and valued, than in the exploration phase, where new hydrocarbon reserves are being sought for future production.

On the day before Christmas Eve in 1969, news reports of the discovery of the Ekofisk field spawned the Norwegian oil and gas adventure. Production from the field began in 1971 and, ever since, the oil and gas industry has had an immense impact on the development of Norwegian society. For the first two decades, Norwegian oil companies concentrated on exploration and production of fields in their own backyard: the North Sea. However, since the early 1990s, they have increasingly focused on international exploration.

Global exploration has introduced the oil and gas industry to a much larger variety of geological scenarios. Many of these new scenarios are harder to analyse than the fields in the North Sea, both with respect to making the discoveries and to producing the located reserves. For example, the presence of salt domes in the Zagros Mountains in Iran and Iraq, like the very thick salt layers in the Gulf of Mexico and along the South Atlantic passive margins outside Brazil, Gabon, and Angola, make discovery and production challenging. Moreover, production in deep waters and horizontal drilling of thin reservoir layers are examples of extreme engineering. Facing such new geoscientific and engineering challenges, the oil and gas industry has been forced to invest massive resources into research and technology development[1].

The Research Collaboration Between Simula and StatoilHydro

In 2004, Simula's former managing director, Morten Dæhlen, initiated contact with Bjørn Rasmussen, who headed Hydro's division for global exploration, to investigate the possibility of a research collaboration between the two companies. The search for common research topics took the form of a pilot study conducted by senior researchers from Hydro's Research Centre in Bergen and from Simula. By the end of 2004, this group had identified several areas of common interest, both from a scientific perspective and in view of practical needs in the industry.

From the beginning it was clear that if the two parties were to enter a collaboration, they would have to set long-term goals for activities rooted in fundamental research questions. In contrast to Norwegian industry in general, Hydro wanted to enter this type of collaboration. This was an unusual and surprising scenario for two reasons: First, Simula had only limited experience with geological and geophysical applications, and the existing experience was related to production rather than exploration. Second, it is generally very difficult to obtain long-term industrial funding in Norway for basic research. Consequently, Hydro's decision presented Simula with at least two challenges: to excel in computational geosciences, and to show the industry that a long-term commitment in the research community can add substantial value to their business.

The report from the pilot study and continued discussions led to the definition and initiation of several projects in 2005. Since then, the number of projects and the intensity in several key activities have increased. From 2006 on, the collaboration with StatoilHydro[2] has generated more than ten per cent of the annual revenue for Simula Research Laboratory. By the end of 2008, about 32 million Norwegian kroner have been invested in this research collaboration. From 2009, the activity is further strengthened.

Main Results

The collaboration has led to strategically important results for StatoilHydro, especially methodology and software tools for the modelling of complicated geological structures on different spatial scales and over geological time. These results include refinement of the Compound Modelling technology and development of the 4D Lithosphere Model, see sections 40.6 and 40.7, respectively. Moreover, recently initiated activities indicate the potential for new approaches to solving important

[1] For an interactive presentation of the Norwegian oil and gas history, see www.statoilhydro.com/no/AboutStatoilHydro/History/AboveAndBelow

[2] The two companies, Hydro and Statoil, merged their petroleum activities in 2007 to form StatoilHydro, one of largest oil and gas corporations in the world. The collaboration between Simula and Hydro was continued and extended in the context of this new company. In the remainder of this paper, we will primarily use the name StatoilHydro, even when referring to activities predating the merger.

problems in the exploration workflow, such as calibration and quality assessment of depositional models, see section 40.4. Promising results are also emerging from ongoing work on a new approach to simulating sand-laden fluid flow and on parallel solutions of three-dimensional models of heat flow and deformation in sedimentary basins, see sections 40.3 and 40.5, respectively.

The work conducted as part of the collaboration ranges from academically oriented basic research to application-driven activities that call both for research and technology development. To maintain a clear financial separation between research-dominated activities and product-oriented development, Simula established the subsidiary Kalkulo AS in 2006. This company is run on commercial terms and conducts the development-oriented tasks in the collaboration. Simula's educational unit, Simula School of Research and Innovation AS, organises the PhD and postdoctoral activities. The coupling between research tasks of an academic nature and technology development has proven to be bidirectional, in that development-oriented activities lead to new PhD and postdoctoral research projects and vice versa. All facets of the collaboration are intrinsically connected and are under the same project management in Simula's Computational Geosciences group. In 2008, this group became part of CBC, a Norwegian Centre of Excellence hosted at Simula.

About this Chapter

After providing an introduction to geological and geophysical preliminaries in section 40.2, we review the research problems addressed by the Simula-StatoilHydro collaboration and their main results. These activities cover modelling of depositional processes in section 40.3; methodologies for well-based calibration of geological models and reliability assessment of their response in section 40.4; simulation of three-dimensional deformation of sedimentary basins in section 40.5; methods and tools for construction of realistic geological models in section 40.6; and advanced software for efficient visual inspection and interaction with time-dependent geological models over multiple spatial scales in section 40.7.

Following this review, we discuss some observations about the collaboration. Although these observations are based on a single case, we believe that there are some fundamental experiences that are widely applicable to collaborative research efforts, even across different industrial sectors.

40.2 Plate Tectonics and Lithospheric Processes

To see how our different research contributions fit into a larger picture, it is necessary to establish some basic concepts and terminology from geology and geophysics. The discussion in the following subsections is by no means exhaustive; rather, it is an attempt to provide a brief sketch of some of the major mechanisms that have formed planet Earth. These mechanisms, and the results they produce, are important factors in oil and gas exploration. The information contained in this section can be found in standard textbooks on plate tectonics and basin analysis, such as [22, 70, 21].

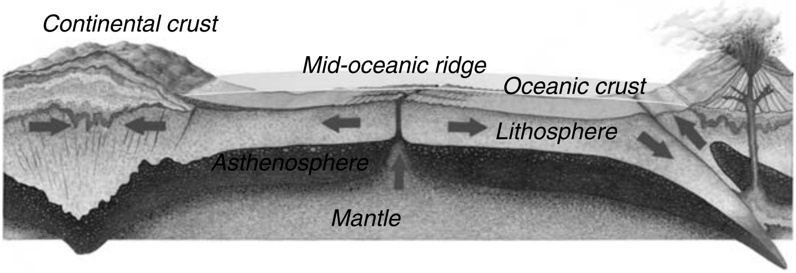


Figure 40.1 A vertical section through the Earth exposing the rigid lithosphere and the flowing asthenosphere. The sketch also illustrates the process of lithospheric stretching, by which the lithosphere is thinned to the point of fracturing or breakup, and molten magma arises from the mantle to form a mid-oceanic ridge. The artwork is courtesy of DK Images [55]. (For the colour version, see figure C.13 on page 650.)

The Lithosphere

The outermost solid shell of the Earth is called the *lithosphere*, which includes the *crust* and the uppermost part of the *mantle*. Below the lithosphere lies the *asthenosphere*, which is the deeper part of the upper mantle, see figure 40.1. In contrast to the mostly rigid lithosphere, the partially molten, weaker, and hotter asthenosphere can, over the geological time scale of several million years, be regarded as a viscoelastic fluid.

The lithosphere is categorised as either oceanic or continental. The oceanic lithosphere is topped by a crust that primarily consists of basaltic rocks and is generally less than 10 km thick, with an average density of about 2.9 g/cm^3. The mantle portion of the oceanic lithosphere thickens with age due to advective heat loss trough the Earth's surface. In comparison, the continental lithosphere and crust are generally older and far more differentiated by geological processes. The crust is mostly composed of granitic rock and is much thicker and less dense than the oceanic type; typically, it is 30–40 kilometres thick, with an average density of 2.8 g/cm^3. The naming of these two types of crust and lithosphere reflects both their genesis and their respective locations: under the large oceans or as the large landmasses commonly known as continents.

Plate Tectonics

The lithosphere is divided into a large number of fragments of varying size that constitute a gigantic puzzle. These puzzle pieces, called *tectonic plates*, slide on top of the slowly flowing asthenosphere. The relative motion between plates may be as fast as 160 mm/year, which is comparable to the speed at which human hair grows.

As tectonic plates move, they cause a time-dependent reconfiguration of the Earth's geometry. This process is associated with both inter- and intra-plate deformation. Continental lithosphere plates are sometimes subject to internal deformation by extensional forces strong enough to tear them apart, thereby forming new plates.

Such full-blown continental breakup allows molten magma to rise from the mantle. This material cools to form new oceanic crust along ocean spreading centres, resulting in mid-oceanic ridges, see figure 40.1. These ridges constitute the longest mountain chains in the world.

While new crust is formed along spreading centres, oceanic lithosphere, being thinner and weaker than the continental lithosphere, is subducted and consumed beneath the latter. Such processes have formed the deep ocean trenches, which are associated with very high earthquake activity. Because the subducted lithosphere is buoyant with respect to the continental mantle, the adjacent continent is lifted, thereby creating mountain ranges such as the Andes mountains in South America. Plate motion will eventually cause pieces of continental lithosphere to collide, forming even larger mountain ranges; an example is when India collided with Eurasia to form the Himalayas.

Tectonic plates may also slide side by side. The resulting fracture along the plate boundary is called a transform fault. Such faults most frequently occur in the oceanic domain. However, the San Andreas fault in California is a prime example of a transform fault that cuts into the continental lithosphere.

Sedimentary Basins

Sedimentary basins play a fundamental role in the search for new petroleum prospects [22]. Such basins have been subjected to prolonged subsidence caused by different mechanisms. Regardless of its history, a basin offers the space required for sediments from nearby regions to accumulate. These sediments are commonly transported into the basin by rivers and gravity-driven mass flows.

Different types of basins are formed in various tectonic settings. When the continental lithosphere is stretched to the point of deformation, the crust will thin and the surface will subside. The North Sea is an example of such an extensional basin that was created by lithospheric stretching about 150 million years ago.

Viewed as a mechanical object, the lithosphere resembles an elastic plate. If a load is placed on top of the lithosphere, it will bend to a degree determined by its composition and the thermal conditions. The load itself can come from different origins. The Hawaiian Islands constitute a special example of a volcanic load causing a deflection of the surrounding oceanic lithosphere. The result is a broad surface depression of the nearby regions. Similar effects are also observed in sedimentary basins as they are loaded by sediments. Over time the basin edges bend downwards towards the basin centre. Basins created by deflection are also found in compressional settings. Here, the load of large mountain ranges will locally cause formation of deep foreland basins. However, due to the relationship between compression/collision and deflection, these basins often are characterised by blocks or sheets of crust and older sediments being stacked on top of each other to form a complicated succession of basin fill.

Deposition and Erosion

The geological process that builds or extends landmasses by sedimentation is referred to as *deposition*. In a sense, *erosion* is the opposite process, namely the displacement of solid material due to the mechanical influence of currents of wind, water, or ice. Over geological time, these processes partly feed each other, as deposited material is eroded and transported for deposition in another location.

Deposition occurs when material of different grain size is transported into sink areas, such as the oceanic basins mentioned above, and the forces that cause sediment transport are outweighed by friction and the weight of the particles. The particles will then settle, building up the layers in a sedimentary basin. Over time, the sediments are buried. As the pressure increases, the material is compacted and cemented to form solid rock of various *lithologic* composition. The increased load due to continued influx and deposition of sediments will cause the lithosphere to sink. On the other hand, if the sedimentary load is reduced by erosion, an uplift will occur. The principle of *isostasy*, which is similar to the buoyancy observed for a body floating in a fluid, seeks to reach a balance between the lithosphere and the asthenosphere. This principle defines an equilibrium for the elevation of the tectonic plates, depending on their thickness and material properties, see [58].

The deposition of sediments in an oceanic basin also depends heavily on sea level, which varies over time due to climatic conditions and the capacity of the oceanic basins. The cyclical changes of global sea level relative to a reference surface is referred to as *eustasy*, see [22] for further information.

Organic material transported with the sediments or already present in a basin can be trapped within the solidified layers. When this type of material is subjected to great pressure under the right thermal conditions over a long period of time, it can be converted into crude oil and natural gas. Four basic conditions must be fulfilled to establish an oil reservoir: (a) source rock that is rich in hydrocarbon must be buried at a depth at which heating from deeper parts of the Earth can help convert it to oil; (b) porous and permeable reservoir rock that is capable of storing the oil must be present; (c) geometrical structures that support accumulation of sediments must be present; and (d) an impermeable seal must be in place to prevent the oil from escaping. In such a reservoir, the geological layer can contain oil, gas, and water as separate phases.

Many different types of depositional processes can give rise to different, possibly distinct, geological features. Thus, proper classification and understanding of the depositional history of a region is an important component in assessing the region's hydrocarbon potential.

40.3 Depositional Modelling

The geology of a sedimentary basin results from thousands of single events that occur over several million years and that influence domains that span tens of kilometres laterally. Due to these temporal and spatial scales, one can not pursue detailed simulations of all of the geological events occurring in the basin over long time periods. Instead, one can try to understand the physics of individual events using simulations that cover very short time intervals and limited computational domains, or one can derive averaged models that allow numerical investigations over larger spans of time and space.

The latter type of models constitutes the class of problems that is commonly referred to by the term *depositional modelling*. Sedimentologists and geomorphologists use several models of this type to simulate the process of sedimentary deposition, e.g., Sedpak [35], Sedsim [29], Dionisos [28], and Sedflux [12]. However, the impact of the current generation of industrial software is quite limited because of the large degrees of uncertainty present in crucial material parameters, such as the diffusive transport coefficients for the different sediments. These parameters are compiled from a number of complex physical phenomena and are therefore difficult to quantify.

In this section, we first focus on a novel method for simulation of particle transport in a sand-laden fluid flow, which may form the basis for models of individual depositional events such as underwater avalanches. Later, in section 40.4, we turn to the topic of how uncertainty in the parameters of a depositional model can be handled. In particular, we discuss a standard model for averaged deposition of sediments in which the parameters governing diffusive transport are subject to uncertainties. We report on recent advances towards automatic calibration of these parameters to observations. Moreover, we discuss an approach to assessing the uncertainty in the computed results. This uncertainty is due to the propagation of the uncertainty associated with the individual process parameters.

Turbidity Currents and Turbidites

Rivers and coastal processes build sandy deltas and shores in shallow waters. For various reasons, these deposits can become unstable and start to slide down slope. The result is an underwater avalanche, whereby the sliding mass is transported into the basin as part of a *turbidity current*. This type of current is a very rapid downhill flow of sediment-laden water that is primarily driven by gravity. The flow is also increased because it has a higher density than the surrounding water masses [22, 2, 20]. Such violent and turbulent high-energy flows can distribute large amounts of clastic sediments[3] over a large area of the basin floor in a very short period of time. Relative to geological time measured in millions of years, a turbidity current lasting for hours or days seems like an instantaneous event.

The sedimentary deposits of turbidity currents are called turbidites [20, 5]. These structures carry visual proof of their history. They show a graded distribution of the sediments; the larger particles settled first and the smaller ones were suspended in the fluid for a longer time. The high energy of these currents can also cause erosion of the underlying layer. This is often reflected in the visual evidence, where the turbidite sequences are incomplete due to more recent turbidity currents eroding the upper part of previously deposited turbidites.

[3] *Clastic* sediments consist of particles from eroded or weathered rocks. A *siliclastic* sediment is a clastic sediment derived from non-carbonate rocks. *Carbonates* are sedimentary rocks such as chalk and limestone that stem from living organisms, such as plankton.

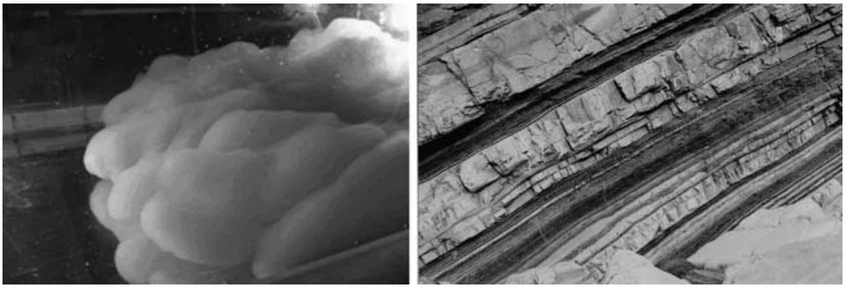


Figure 40.2 Left: The image shows a snapshot of a turbidity current created in a tank experiment (Photo: Jerome Neufeld [56]). Right: Repeated turbidity currents have created a stack of turbidites, here exposed in an outcrop in Cornwall, UK (Photo: Kevin Walsh [57]).

Turbidity currents and debris flows[4] are the most important gravity flows that transport sediments into the deep sea. Stratigraphic records from deep water regions usually exhibit proof of such flow types, and they are often the basis of petroleum reservoirs [20]. Proper understanding of these sediment bodies and their history is therefore an important aspect of oil and gas exploration.

Simulation of Particle Transport in Sand-laden Fluids

Underwater gravity flows of sediments can be investigated using numerical models. In traditional flow simulation, these processes are modelled by continuum physics and typically are described by a set of partial differential equations (PDEs). The resulting equations can then be solved using numerical methods from the field of computational fluid dynamics (CFD).

As an alternative to continuum-based CFD methods, particle-based methods can be used to address this type of flow problem. Such methods formulate a set of physical rules that control the local interaction between particles. Several scientific studies have used particle-based approaches to sedimentation problems, although they seem not to have been widely applied to geological deposition. Relevant examples include sedimentation in pipes [7], transport of airborne contaminants [8], and snow avalanches [9]. One particular application relevant to geological processes is the use of cellular automata to model turbidity currents, see Salles et al. [51].

In a parameter-based model, the particles are not necessarily physical entities. In fact, one can define rules acting on lumped particles that are composed of smaller, physical particles. In this case, the particle-based rules will cause splitting and aggregation of the lumped particles as time passes and the sediments are transported through the computational domain. Most likely, a fully Lagrangian approach, in which all individual physical particles are traced, would be the most accurate method. However, the lumped particle approach presented here offers a reasonable trade-off between accuracy and computational feasibility.

[4] *Debris flows* are laminar flows driven by gravity.

In the context of a turbidity current, one can consider models consisting of two essential parts: the submodel describing the evolution of the fluid flow itself, and the submodel describing the transport of sediment particles in the given fluid flow field. In an ongoing PhD project, we are currently investigating a novel particle-based model for sediment transport, see al-Khayat et al. [48]. In this particular work, the velocity field describing the fluid flow is assumed to be known; that is, this field could be computed by a continuum-based Navier-Stokes model [18], or, as discussed in an earlier work [43], it could be the result of a Lattice-Boltzmann computation. On the other hand, the mass transport is modelled in a hybrid continuum-particle fashion. To the best of our knowledge, this model is a novel contribution. It benefits from simple problem descriptions with a close connection between observable properties and the model parameters. Moreover, the particle-based nature of the physical problem makes it reasonable to adopt a particle-oriented computational procedure. This approach seems to lend itself more naturally to the modelling of deposition and erosion than methods based on the continuum principle. The suggested method can correctly model physical processes covered by traditional PDE-based formulations, such as convection and diffusion. In addition, it is designed to handle dispersion, which is difficult to model correctly using continuum methods.

The time-dependent evolution of a lumped particle suspended in the fluid flow is described by the Bassinet-Boussinesq-Oseen (BBO) equation, see [6]. This equation can take different forms depending on which physical effects are included. Currently, we use the simple formulation

$$\frac{d\mathbf{v}_p}{dt} = \underbrace{\frac{1}{\tau_p}(\mathbf{v}_p - \mathbf{u})}_{\text{Stokes drag}} + \underbrace{\left(1 - \frac{\rho_f}{\rho_p}\right)\mathbf{g}}_{\text{buoyancy}}. \tag{40.1}$$

Here, $\mathbf{v}_p$ denotes the particle's unknown velocity, while the other entities are known. In particular, we know the particle's relaxation time τ_p, the flow velocity of fluid $\mathbf{u}$, the acceleration $\mathbf{g}$ due to gravity, particle density ρ_p, and fluid density ρ_f. Equation (40.1) is solved numerically with a finite difference scheme.

The computational domain in two- or three-dimensional space is discretised in terms of a regular grid consisting of rectangular cells. When a lumped particle has been transported to a given cell in the grid, the proposed algorithm shifts from continuum to particle mode. The lumped particle is split into a set of quasi-particles, and each quasi-particle is dispersed in a predefined number of directions, each with its decomposed velocity, see figure 40.3 (left). The quasi-particles present in each cell are recombined to form a new lumped particle.

Because Lagrangian particle transport must be mapped to the fixed grid used by the underlying fluid flow solver, special precautions must be taken to compensate for the truncation of the positional information. For this reason, the algorithm keeps track of the offset between the true position of each lumped particle and the centroid of the grid cell hosting the particle, see figure 40.3 (right). In this way, computation of the next transport step will be based on the correct particle position. Systematic experiments with simple test problems have shown that this new method is consis-

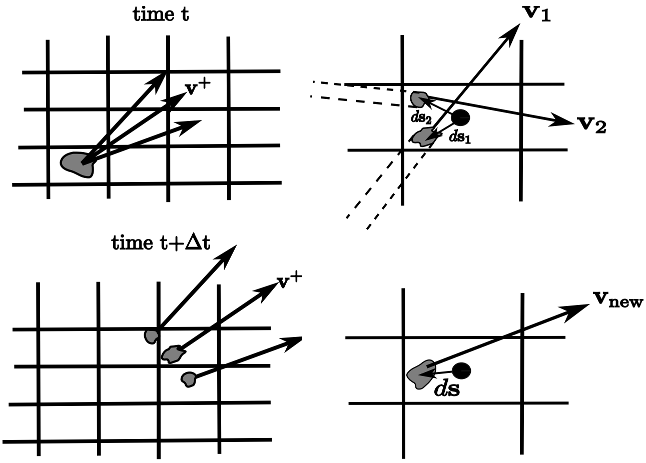


Figure 40.3 Left: A lumped particle is split and dispersed along a predefined number of symmetric directions at time t. At the next time step $t + \Delta t$, each quasi-particle has reached another grid cell determined by its individual velocity vector. All quasi-particles present in the same cell will recombine to form a new lumped particle. Right: To compute the accumulated particle motions accurately, the algorithm records the offset $d\mathbf{s}$ between the true position of the lumped particle and the centroid of the grid cell. This offset is computed by averaging the offsets for each quasi-particle that contributes to the lumped particle. The illustrations are courtesy of Omar al-Khayat et al. [48].

tent with the physics of the system and with accepted continuum-based methods. By proper initialisation of the directions used for the dispersion, one can also reproduce continuum-based simulations of pure diffusive processes. Further details are available in [48]. Given the confirmation of the new method's strong ability to reproduce the physics of the system, the next phase of this work will concentrate on the deposition and erosion of particles.

40.4 Calibration and Reliability of Geological Models

Most geological models rely on a combination of observations of and assumptions about the underlying processes. Both the observations, which typically take the form of interpreted seismics and well measurements, and the geologists' assumptions are subject to considerable levels of uncertainty. In many cases, the inherent uncertainty causes more or less random guesses for the values of critical parameters in the model.

When uncertain data are input into a mathematical model, the results obtained from the model will also be uncertain. Whether the uncertainty has been amplified or reduced depends on the physics of the particular model. Without knowledge of the uncertainty in the model's input parameters and how this uncertainty affects the computed result, the quality of the simulation is not well defined.

Clearly, mathematical models would be more valuable if they were complemented with procedures that could calibrate the model parameters to accurate observations, or at least provide measures of the uncertainty embedded in the computed results. In this section we discuss such procedures in the context of a depositional model. This model is subject to large degrees of uncertainty in the material-dependent parameters that govern the transport.

Averaged Effects of Deposition

Simulation of the depositional history of a wide area over long periods of time is feasible only when using averaged models. Usually, averaged models for the transport, deposition, and erosion of siliclastic sediments are taken to be diffusion dominated, as in Demostrat [26, 27]. The same principle is also present in the industrial simulator Dionisos [28], although other transport effects have been added (e.g., wave-induced processes). Some models include other transport mechanisms, such as the dispersive and advective terms in Simsafadim [34]. However, the diffusion-oriented Dionisos model is the predominant choice in the oil and gas industry, possibly followed by Sedsim [29].

All of the different variants of models referred to above have a similar structure, in that the kernel of these models is a system of PDEs. The unknowns in this system are typically the thickness $h = h(x,y,t)$ of the layer being deposited and the corresponding volume fractions $f_i = f_i(x,y,t)$ of each of the involved sediment types. The different sediments give rise to different lithologies. A slightly simplified version of the Demostrat model [26], adapted from two to n lithologies, reads

$$\begin{aligned} \frac{\partial h}{\partial t} &= \sum_{i=1}^{n-1} \frac{1}{c_i} \nabla \cdot (k_i f_i \nabla h) + \frac{1}{c_n} \nabla \cdot (k_n (1 - \sum_{i=1}^{n-1} f_i) \nabla h) \\ \frac{\partial f_i}{\partial t} + f_i \frac{\partial h}{\partial t} &= \frac{1}{c_i} \nabla \cdot (k_i f_i \nabla h), \end{aligned} \tag{40.2}$$

for $i = 1, 2, \ldots, n-1$, where (x,y) is the position in a basin B, and $t>0$. Here, c_i and k_i denote the concentration and the diffusive transport coefficient for the ith lithology, respectively. Because $\sum_{i=1}^{n} f_i = 1$, we have $f_n = 1 - \sum_{i=1}^{n-1} f_i$. Consequently, there is only need for $n-1$ dynamic equations for the unknown fractions $f_1, f_2, \ldots, f_{n-1}$.

Depositional models that combine siliclastic sediments and carbonates also exist. These models contain a system of PDEs similar to (40.2). However, in contrast to (40.2), these PDEs are then extended with a system of ordinary differential equations that model the growth of the organic species that produce the carbonates, see [34].

From the very beginning of the collaboration between Simula and StatoilHydro, considerable concern has been expressed regarding the use of depositional models. The geologists agree that it is important to understand the depositional history of a prospective oil or gas field. However, hardly anyone will trust the results generated by a depositional model because of the lack of confidence in the choice of the parameters governing the diffusive transport i.e., the values k_i in (40.2). The values of such parameters are based on a combination of several complex geological and physical phenomena. It is therefore very difficult to determine accurate parameter values with a high level of confidence.

Consider the dual-lithology variant of (40.2) with $c_1 = c_2 = 1$,

$$\begin{aligned} \frac{\partial h}{\partial t} &= \nabla \cdot (\alpha s \nabla h) + \nabla \cdot (\beta(1-s)\nabla h) \\ \frac{\partial s}{\partial t} + s\frac{\partial h}{\partial t} &= \nabla \cdot (\alpha s \nabla h), \end{aligned} \tag{40.3}$$

where $s = f_1$, $\alpha = k_1$, and $\beta = k_2$. Here, the lithologies involved are typically sand and mud. Using the simplified model (40.3), we have addressed the uncertainty in the diffusion coefficients α and β in two different ways. As discussed below, we have developed a procedure that allows us to calibrate the diffusion coefficients to observations in specific points. For a real application, these points would correspond to wells. Moreover, we have developed a stochastic approach to depositional modelling in that α and β are considered to be stochastic variables. Consequently, the output of the simulation are the *expected* values[5] of h and s and measures of the variance of the computed values. This approach is explored further starting on page 567.

Calibration of Depositional Models

Over the past years, researchers at Simula have been working on *inverse problems*. Based on electrocardiographic (ECG) measurements, they seek to estimate the size and location of damaged heart tissue due to ischemia[6], see chapter 22. There are similarities between this medical application and the problem of identifying the diffusion coefficients in a depositional model, such that the solutions obtained from the model will match physical observations in selected well positions. Recently, we have investigated the possibility of developing a self-calibrating depositional model using the same approach as that used in the heart simulations.

The simplified depositional model (40.3) is a *forward problem*, which maps the transport parameters (α and β) to the computed output. As pointed out by Imhof and Sharma [41], it is impractical or even infeasible to recover the optimal transport coefficients by a direct inversion. Instead, our approach is to use the forward transient evolution of the sediment layers in one or more small regions to *calibrate* α and β. Typically, these regions will be wells, for which we can extract reasonably accurate observations. By inserting the calibrated parameters into the depositional model, we can simulate the historical evolution of the complete basin; that is, the evolution of the entire basin composition is determined by local data from the wells.

Minimisation of the Output Functional. Assume that we have observed a particular sand fraction $\widetilde{s}$ and layer thickness $\widetilde{h}$ in the wells. To calibrate the parameter values to these observations, we define the output functional

$$J(\alpha,\beta) = \frac{1}{|W|}\int_0^T \int_W (\widetilde{s}-s)^2 + (\widetilde{h}-h)^2. \tag{40.4}$$

[5] The expected value of a stochastic variable is also referred to as the mean value.

[6] Ischemia is an early stage of heart infarction.

Here, W denotes the union of all well domains in question, and $|W|$ denotes the area of W. The functional J is a measure of the deviation between the observations $\tilde{s}$ and $\tilde{h}$ and the output s and h generated by the model at the well positions. Therefore, we want to find the values of the parameters α and β that minimise the value of J, see Schroll [31]. This minimisation problem is constrained by the PDEs in (40.3) because J depends on α and β only via the solutions s and h.

When minimising a function, many computational algorithms require evaluations of the gradient of the function. For this particular problem, we use the Landweber algorithm, which is an iterative steepest descent method [30]. After solving the PDE system (40.3), this algorithm requires us to compute the next search direction in the parameter space,

$$\mathbf{d}^k = -\frac{\nabla J(\mathbf{p}^k)}{||\nabla J(\mathbf{p}^k)||}. \tag{40.5}$$

For our specific problem, $\mathbf{p}^k = (\alpha^k, \beta^k)^T$ is the current estimate for the optimal parameter values. Once $\mathbf{d}^k$ is computed, the estimate is updated by taking a step in the search direction, $\mathbf{p}^{k+1} = \mathbf{p}^k + \gamma^k \mathbf{d}^k$, where γ^k is determined by a line search, see [31].

The evaluation of the gradient in (40.5) calls for differentiation of J with respect to all of the involved parameters. This information is usually not available in analytical form, and numerical differentiation is infeasible for non-trivial problems. In fact, numerical differentiation would require the solution of $m+1$ nonlinear PDE systems of the form (40.3), where m is the number of parameters we want to calibrate. However, the required gradient information can be extracted from the solution of an auxiliary system of PDEs, namely the *dual problem* of (40.3). This dual system, which is of the same complexity as the forward problem, can be solved stably backwards in time. Thus, even for calibration of only a few parameters, there is a significant reward in applying the dual formulation instead of numerical differentiation. The details of the dual problem that matches the depositional model (40.3) is given in [31]. An obvious disadvantage of the described approach is that one needs intimate knowledge of the forward problem in order to define the dual system. Thus, this approach can not be combined with industrial applications for depositional modelling without having access to all details of the underlying mathematical model. However, one can eliminate this disadvantage by replacing the Landweber iterations with a derivative-free optimisation method, see [32, 54]. Such an approach eliminates the need to formulate a dual problem, but it may introduce efficiency and stability issues due to different convergence properties.

Numerical Inspection of the Output Functional. Let us briefly inspect the behaviour of the functional J. One would assume that J has an irregular behaviour, thus it is surprising to discover that J is convex and has a unique minimum. To illustrate this property, consider the depositional model (40.3) over a basin $B = [0,1]^2$ with discontinuous diffusion coefficients:

$$\alpha = \begin{cases} \alpha_1 & x \geq 1/2 \\ \alpha_2 & x < 1/2 \end{cases}, \quad \beta = \begin{cases} \beta_1 & x \geq 1/2 \\ \beta_2 & x < 1/2 \end{cases}.$$

This definition causes our minimisation problem to be four dimensional. We let $(\alpha_1, \alpha_2) = (\beta_1, \beta_2) = (0.8, 0.8)$ and ran the model (40.3) to create a reference solution. This reference solution was then perturbed with five per cent random noise to form the observations needed to define the functional J in (40.4). That is, the perturbations of s and h were used as observations $\widetilde{s}$ and $\widetilde{h}$ in the evaluation of J, given an appropriate choice of the well domain W.

For the first experiment we fixed $\beta_1 = \beta_2 = 0.8$. The contours depicted in figure 40.4 (top) clearly show the convex nature of J. We next tried to minimise J with this particular choice of $\widetilde{s}$ and $\widetilde{h}$. Ideally, the minimisation should converge to a point reasonably close to the parameter values α and β that were used to compute the underlying reference solution. We selected starting values for α_1 and α_2 and ran the Landweber algorithm until convergence was reached in the sense that $||\nabla J||$ became less than a given tolerance. The dotted paths in figure 40.4 (top) shows the sequences of iterative approximations of α_1 and α_2 (blue markers) obtained for four specific choices of starting values. In each case, the iterations converged to an approximate minimum (green markers, one at the end of each path). As expected, the computed minima for the different starting values are all located in the neighbourhood of the values used to compute the underlying reference solution (red marker). Similar results are obtained if one fixes two other parameters and seeks the optimal choice for the remaining ones.

To test this approach more thoroughly, we conducted another experiment in which we minimised J when all four parameters were allowed to change. Again, we added five per cent random noise to the well data, $\widetilde{s}$ and $\widetilde{h}$, before repeating the minimisation procedure for different starting values. Figure 40.4 (bottom) shows the calibrated parameter values for 50 data realisations. Each blue dot indicates the computed values of α_1 and α_2 for one realisation, while the corresponding estimates of β_1 and β_2 are shown in green. As expected, the averages of the optimised parameter values shown as red dots approximated the reference parameters $(\alpha_1, \alpha_2) = (0.6, 1.0)$ and $(\beta_1, \beta_2) = (1.0, 0.6)$ quite well.

The results obtained so far show that the inverse approach to automatic calibration of a depositional model with two lithologies is feasible. However, to what degree the convexity of the error functional J will hold when one increases the number of parameters to calibrate remains an open question. Future work will investigate this property by focusing on extensions to multiple geological layers and multiple lithologies. Moreover, we will consider extensions of the depositional model to include effects that at are not present in the PDE system. In particular, this includes compaction of sediments, tectonic and isostatic subsidence, and eustatic variations of the transport coefficients.

Assessing the Reliability of Depositional Models

Models involving parameters that are subject to uncertainty may also benefit from a *stochastic* approach, in which the questionable parameters are considered to be random variables. Under suitable assumptions about continuity, each parameter is assigned a probability density function, which provides the parameter with an ex-

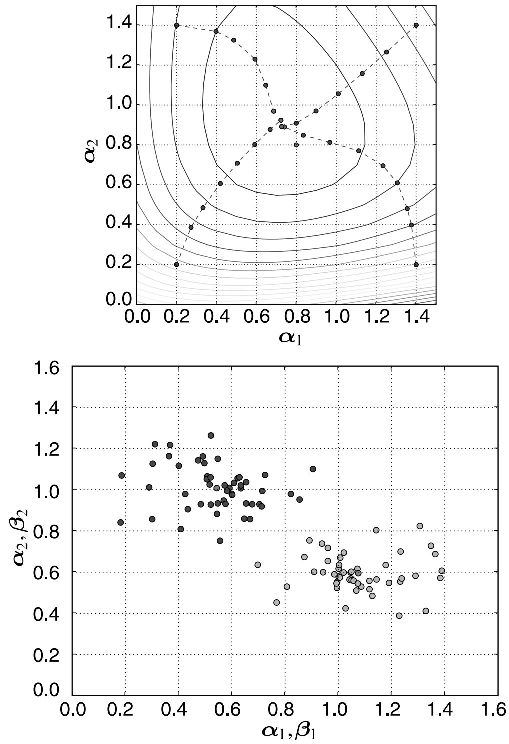


Figure 40.4 Top: Using the reference solution with five per cent noise added, the green dots indicate the paths of the Landweber iterates in the α_1-α_2 parameter space for four different starting values. In all cases the iterates converge towards a minimum. This minimum is located in the neighbourhood of the reference parameters $(\alpha_1, \alpha_2) = (0.8, 0.8)$, which are marked with a red dot. Bottom: The four parameters have been calibrated for 50 cases in which the reference solution has been perturbed by 5% random noise. The averages (red dots) approximate the underlying reference parameters $(\alpha_1, \alpha_2) = (0.6, 1.0)$, $(\beta_1, \beta_2) = (1.0, 0.6)$ quite well. The blue dots indicate computed values for α_1 and α_2, whereas the green dots refer to values of β_1 and β_2. The illustrations are courtesy of Hans-Joachim Schroll [31]. (For the colour version, see figure C.14 on page 651.)

pected value μ and a standard deviation σ. Replacing the deterministic parameters in a traditional model with the corresponding random variables produces a stochastic model. The output, or *response*, of the model then becomes one or more random variables with their own probability densities. Running a large number of simulations that reflect the random variation in the stochastic parameters serving as input provides an estimate of the potential outcome of the model. Each simulation is an execution of a deterministic model, also referred to as a realisation of the stochastic model. To provide a good estimate of the response, it is necessary to sample the parameter space in such a way that the possible output space is satisfactorily spanned. In its simplest form, such a procedure could be a naive implementation of the Monte Carlo method [15]. However, this approach can be extremely expensive in terms of computing time and may be practically infeasible. It then becomes necessary to explore alternative approaches that can reduce the number of parameter samples needed to estimate the variation of the model's response. For a general discussion of stochastic modelling, see [13, 14].

Advanced mathematical tools seem to be more prominent in oil and gas production than in exploration. In terms of handling uncertainty in geological process models, stochastic methods have been applied successfully to the porous media flow problems typically found in reservoir simulation [33]. In fact, this observation spawned our research on how stochastic methods can be applied to depositional models. From the discussion in the beginning of this section, one approach would be to consider the diffusion coefficients in (40.2) or similar models as random variables. Thus, after an introduction to the basic methodology, we present some results of this approach.

The Probabilistic Collocation Method. Several recently developed methods for stochastic simulations are based on the Karhunen-Loève (KL) expansion, which represents a stochastic process as an infinite linear combination of orthogonal functions [10]. Conceptually, the KL expansion is analogous to the decomposition of a compound signal into its respective frequency components as in a Fourier series expansion. The *Polynomial Chaos Expansion* is a computationally attractive alternative to the KL expansion, see [17]. The polynomial chaos expansion of the random variable u in terms of the parameters $\mathbf{p} = (p_1, p_2, \dots, p_N)$ would be

$$\begin{aligned} u(\mathbf{x}, t; \mathbf{p}) = a_0(\mathbf{x}, t) Q_0 &+ \sum_{i=1}^{\infty} a_i(\mathbf{x}, t) Q_1(p_i) \\ &+ \sum_{i=1}^{\infty} \sum_{j=1}^{i} a_{i,j}(\mathbf{x}, t) Q_2(p_i, p_j) \\ &+ \sum_{i=1}^{\infty} \sum_{j=1}^{i} \sum_{k=1}^{j} a_{i,j,k}(\mathbf{x}, t) Q_3(p_i, p_j, p_k) + \cdots \end{aligned} \tag{40.6}$$

Here, the multidimensional polynomials Q_d have degree d and form an orthogonal basis. The parameters p_i are the random variables that represent the uncertain parameters in the deterministic PDE model. By assigning different types of PDEs to these parameters, different classes of basis polynomials will be suitable. If necessary, these polynomials can be constructed by recurrence relations or similar procedures [16]. Alternatively, one may find use for a classical type of orthogonal polynomials. In particular, when the parameters have a standard normal (Gaussian) distribution, Q_d will be the orthogonal Hermite polynomial[7] of degree d, i.e., $Q_d(\mathbf{p}) = H_d(\mathbf{p})$. Any normal distribution can be transformed to the Gaussian case, which means that the Hermite polynomials can be applied.

[7] Be aware that there are two definitions of Hermite polynomials, one commonly used in probability theory and one often applied in physics. These two types of polynomials are related through scaling. The discussion of polynomial chaos expansions assumes the use of the probabilistic variant of the polynomials.

Consider a PDE of the form $L(u) = f$, which contains N uncertain parameters. Viewing these parameters as random variables p_i, we apply the polynomial chaos expansion of the stochastic solution u as defined in (40.6). For computational purposes, we use a truncated expansion of degree m, which contains $P = (N+m)!/(N!m!)$ terms. For $m = 2$ and $Q_d = H_d$, we get:

$$\begin{aligned}\hat{u}(\mathbf{x},t;\mathbf{p}) = a_0(\mathbf{x},t) + \sum_{i=1}^{N} a_i(\mathbf{x},t)p_i \\ + \sum_{i=1}^{N} a_{i,i}(\mathbf{x},t)(p_i^2 - 1) \\ + \sum_{i=1}^{N-1}\sum_{j>1}^{N} a_{i,j}(\mathbf{x},t)p_i p_j \\ = \sum_{j=1}^{P} c_j(\mathbf{x},t)\Psi_j(\mathbf{p}).\end{aligned} \tag{40.7}$$

Inserting $\hat{u}(\mathbf{x},t)$ into the PDE to form the residual $R(\hat{u}(\mathbf{x},t)) = L(\hat{u}(\mathbf{x},t)) - f$, the coefficients c_j can be determined by the weighted residual method using either a Galerkin-type argument or collocation. The first option defines the Polynomial Chaos Expansion (PCE) method, whereas the latter leads to the *Probabilistic Collocation Method* (PCM), see [1]. In contrast to the PCE method that leads to a system of coupled equations, the PCM approach gives a set of independent deterministic PDEs of the same form as the original problem. The PCM is generally believed to be more efficient than the PCE method, see [3].

When applying the PCM, we need to select P collocation points $\mathbf{p}_j$ that will be used to compute the unknown coefficients c_j. More precisely, the deterministic model is evaluated by running the simulator for each of these parameter sets $\mathbf{p}_j$ and recording the response values $u_j, j = 1,2,\ldots,P$. Given the fixed collocation points and the associated responses, we can compute the coefficients c_j by solving a $P \times P$ linear system of algebraic equations.

Following the same principle as in the construction of Gaussian quadrature rules for numerical integration [19], the collocation points are usually based on the roots of the next higher degree orthogonal polynomial. In our example, this next polynomial is the Hermite polynomial $H_3(\xi) = \xi^3 - \xi$. If we consider a problem with $N = 6$ uncertain parameters, we need $P = (6+2)!/(6!2!) = 28$ collocation points to determine the 28 coefficients c_j. However, because H_3 has three roots, there are $3^6 = 729$ possible choices. Usually, one selects the points that correspond to the highest probability. Alternatively, one may pick random points among the possible candidates.

Different parameters p_i can have different types of probability distributions, which will lead to different classes of basis polynomials in the polynomial chaos expansion (40.6). Moreover, the PCM is non-intrusive in the sense that the equations to be solved for each collocation point will be the same as those in the original deterministic model. That is, we can view the underlying model as a black box, and the PCM will work as long as we can evaluate the model for a given input $\mathbf{p}_j$. Consequently, the

PCM approach can be combined with existing codes, and it can be easily parallelised because the P evaluations of the model response are independent.

The polynomial response function derived by PCM or similar methods can be used as an approximation of the underlying model, which in turn can be used for the uncertainty analysis. If the results from this analysis are to be meaningful in the context of the original model, it is important that the error introduced by the approximation is acceptably small. Similar to the use of Gaussian quadrature rules, one may estimate the error of the polynomial approximation $\hat{u}$ by evaluating the model for additional collocation points. These additional points are obtained from the roots of higher order basis polynomials and will lead to more runs of the underlying simulator. If the error is too large, the polynomial response function $\hat{u}$ should be extended with higher-order terms, and the error analysis should be repeated.

Once the response of the model is represented by a polynomial chaos expansion, we can estimate the behaviour of the model by statistical sampling. Obviously, it is much less demanding to sample a truncated version of the polynomial (40.6) than to sample the original PDE model, which would be the case for a traditional Monte Carlo simulation. In particular, the basic statistical moments are directly accessible because

$$\begin{aligned} \mathrm{E}[u(\mathbf{x},t)] &= c_1(\mathbf{x},t), \\ \mathrm{Var}[u(\mathbf{x},t)] &= \sum_{j=2}^{P} (c_j(\mathbf{x},t))^2 \mathrm{E}[\Psi_j^2], \end{aligned} \tag{40.8}$$

where Ψ_j is given by (40.7).

Application of the PCM to Depositional Models. Considering depositional models of a form similar to (40.2), it would be natural to look at the coefficients in the dynamic equations as random variables. However, one could also apply the same principle to other conditions that influence the behaviour of the system, such as the initial geometry on which sediments were deposited (paleobathymetry), the time-varying influx of sediments, and the initial composition of sediments in the basin. If the model is extended to incorporate effects not covered by the PDEs, such as eustasy and compaction, parameters governing these effects could be interpreted as random variables. The PCM approach discussed above could be applicable to any of these types of model parameters. However, our present study focused on the diffusion coefficients k_i, or, in the two-lithology case, α and β.

When representing the diffusion coefficients as random variables, the computed sediment thickness h and volume fractions f_i also become random variables. The goal of the computation is to estimate the corresponding expected values $\mathrm{E}[h]$, $\mathrm{E}[f_i]$ and the variances $\mathrm{Var}[h]$, $\mathrm{Var}[f_i]$ as given by (40.8) at different time steps.

We applied the PCM methodology to several implementations of diffusion-based depositional models, see Clark et al. [50]. Initially, we used an in-house implementation of the basic two-lithology model (40.3) defined over one- and two-dimensional computational domains[8]. We refined this implementation into a fully parallelised code, which is particularly suitable for experimentation with higher-order basis polynomials, high resolution grids, or other method-related settings that increase the computational burden. For instance, simple experiments with the one-dimensional version of this code give information on the expected sediment thickness at different time steps, including the standard deviation, see figure 40.5. For this particular example there is only one lithology and no influx of sediments. The PCM approach with a linear polynomial suggests that the thickness varies proportionally with both the slope and the mean diffusion coefficient. Points with little variance experience little change in the mean height. Therefore, the standard deviation is small at $x = 0$ for $t = 0.0212$, as the major diffusive effects occur at $x = 0.18$ where there is a sharp change in the initial hill slope. As t increases and the slope is eroded, the uncertainty of the computed thickness h increases for points corresponding to the greatest change of $\mathrm{E}[h]$. This behaviour, which is observed for instance at $x = 0$, occurs because the diffusion coefficient is greatest at these points.

For the PCM approach to be valuable for industrial problems, it must be combined with already established simulators. As a first test of the applicability of this solution strategy, we combined a Python implementation of the PCM with the existing FORTRAN-based simulator Demostrat [27]. By writing interface scripts that can edit input files for Demostrat and extract the computed solution from the generated output files, we were able to instrument the Demostrat model without any changes to the original code. Figure 40.6 shows expected values and standard deviations for the thickness h and the volume fraction of sand s when the PCM was applied to Demostrat runs of one of our test cases. In this particular case, sand flowed in from the right ($x = 200$) and was blocked by the bank at $x = 120$. As the bank eroded, a certain volume of sand migrated to the left. However, the trough at $x = 50$ was then already mostly filled. As indicated by the plots of standard deviations, the uncertainty in the basin morphology was greatest at the right and lowest at the left. This situation occurred because the infill came from the right. For this case, we assigned probability densities such that the diffusion coefficient for sand (shale) varied linearly from 20 (10) to 100 across the basin from left to right, with a standard deviation of 0.1.

Recently, we embedded the Dionisos simulator into our PCM software framework [50], see figure 40.7. We expect to try other industrial simulators in the near future. These simulators contain a large number of parameters, and most are subject to uncertainty. To keep the computational complexity at a reasonable level, it will be important to use geological knowledge and intuition to carefully select which parameters to regard as random variables. The complexity of the parameter space in these simulators is extreme because the diffusion coefficients depend on a number of different conditions, such as sediment grain size, water depth, climate and vegetation, being in a marine or non-marine environment, or being in a fluvial or non-fluvial environment. All of these conditions are, by themselves, uncertain parameters. This

[8] In the literature on depositional modelling, the terms two- and three-dimensional models are used. This terminology combines the dimensionality of the computational domain *and* the z (thickness) direction.

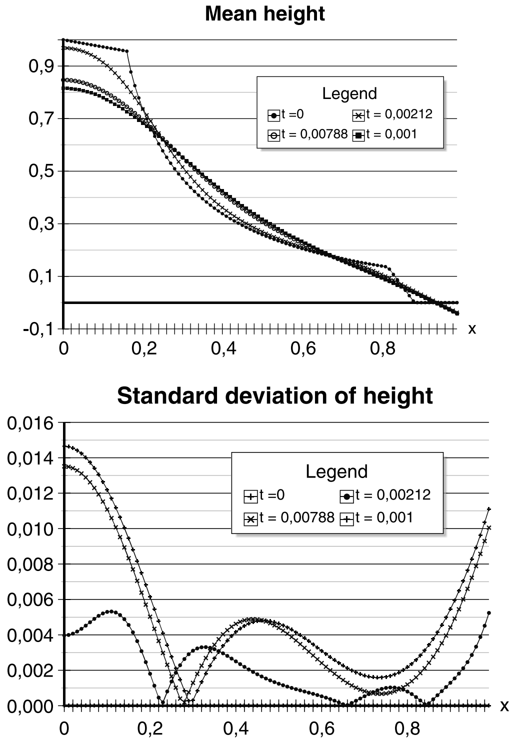


Figure 40.5 Expected values (top) and standard deviation (bottom) of the sediment thickness *h* at different time steps. The illustrations are courtesy of Stuart Clark et al. [50].

picture gets even more complicated when one takes into account effects such as compaction and eustasy.

Finally, several of the conditions that influence a depositional model vary with geological time. In the context of stochastic modelling, one challenge will to be assign reasonable probability density distributions for the diffusion coefficients and other relevant parameters. It is likely that one can construct a gallery of probability density functions for different specific instances of the different parameters and tie these distributions to different problem scenarios. The geologist would then have to decide which scenario best fits the current project, thereby indirectly selecting a set of distributions that define the specific stochastic problem.

40.5 Deformation and Heat Flow in Sedimentary Basins

As discussed in section 40.2, the processes leading to the presence of oil and gas in a sedimentary basin are heavily influenced by external forces and thermal conditions over long periods of time. Therefore, it is essential to investigate the deformation and heat flow as an integral part of the *basin analysis* [22] conducted during the

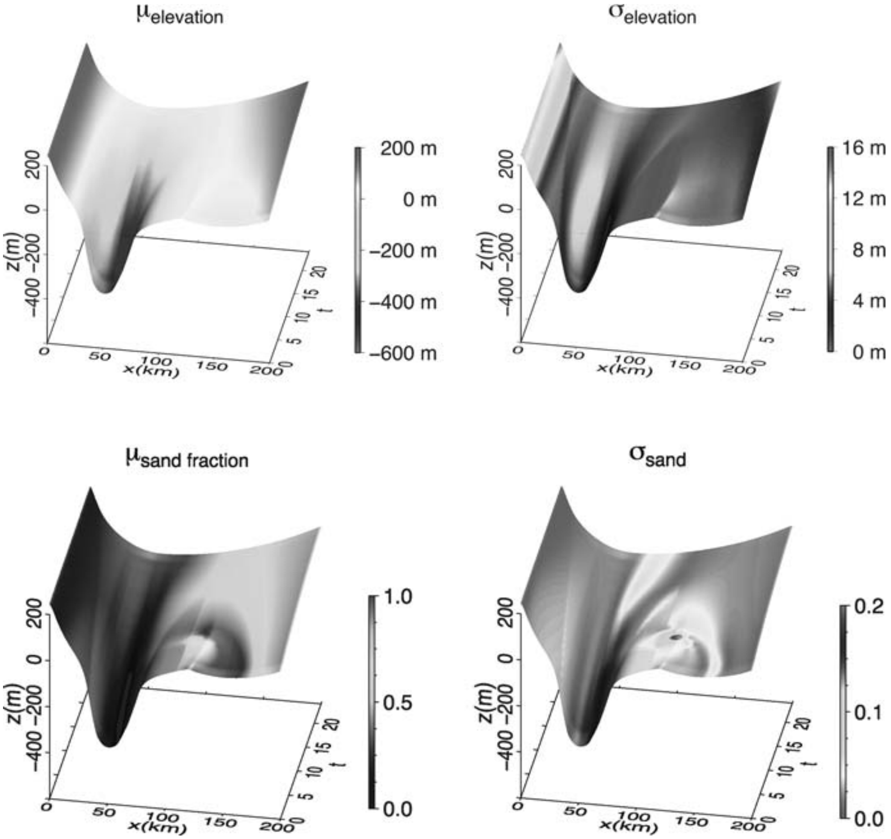


Figure 40.6 Top, left to right: Expected values and the standard deviation of the thickness (*h*) in a Demostrat-based model. Bottom, left to right: Expected values and the standard deviation of the volume fraction of sand (*s*) in the same model. For each figure, the result is shown as a coloured map draped on top of a surface depicting the expected thickness varying with time. The plots have one spatial and one temporal axis, meaning that the model is one-dimensional. The illustrations are courtesy of Stuart Clark et al. [50].

exploration phase. This type of analysis requires numerical simulation of these fundamental processes.

High-quality estimates of pressure conditions and fracturing in geological structures are important results of basin analysis. The traditional software products used for this type of analysis neglect the computation of lateral stress effects in the sediment packages. Instead, they usually apply simple one-dimensional compaction rules in the vertical direction only. This approach is based on Terzhagi's law of compaction, which relates the horizontal stress components to the overburden sedimentary load acting vertically. The rationale behind this design is that of computational efficiency; accurate solution of mathematical models for three-dimensional deformation is extremely time consuming. Nevertheless, several geological scenarios exist in

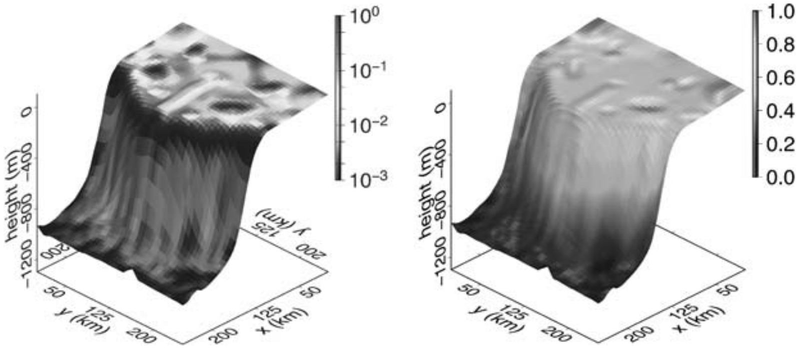


Figure 40.7 Expected values (left) and standard deviation (right) of the volume fraction of sand in a Dionisos model for a prograding delta. An even mix of sand, shale and silt is injected into the model at $x = 0$ km, $y = 125$ km and builds a delta out onto the continental shelf. In the mean outcome, sand remains on the contintental shelf and slope, although certain model outcomes allow for significant sand volumes on the lower slope, as is shown on the right. Together these two pictures capture the probability distribution of model outcomes. The illustrations are courtesy of Stuart Clark et al. [50]. (For the colour version, see figure C.15 on page 652.)

which the simplified procedures may be too weak. For instance, the simplified approach makes it impossible to account for the major influence that extensional or compressional tectonic features or salt doming can exert on basin processes.

The Three-dimensional Porothermoelastic Model

Kjeldstad et al. [52] present a model that couples heat flow, fluid pressure, and deformation of solids for the two-dimensional case of a vertical basin section. In particular, this model was used for the numerical investigation of magmatic *sill intrusions*[9] in a sedimentary basin. More recently, this model was extended to three-dimensional space and to have stronger couplings between the PDEs. It has been implemented from scratch as a Diffpack [53] simulator. The numerical simulator based on this porothermoelastic basin model can compute the stress distributions in the basin under different loading regimes. Moreover, the simulator is designed to read native models generated by the basin analysis software Petromod[10], which is commonly used in the oil and gas industry. Our interface tools can import the basin geometry as well as all relevant data sets and parameters from a Petromod model. This information can then be utilised in the specialised simulator to compute the stress.

Traditional computations in basin analysis cover several million years of geological evolution. Due to the complexity of the stress calculations, our detailed simu-

[9] Sill intrusions are horizontal sheets of magmatic material, which typically show evidence of strong heating.

[10] See information at www.ies.de.

lations should typically be performed for shorter time intervals, e.g., up to tens of thousands of years. The workflow begins with conventional Petromod runs. At the geological time at which one expects strong lateral stress effects to build up, the Petromod model is exported to the specialised three-dimensional simulator for a detailed, short-term simulation.

The mathematical model for the porothermoelastic basin model is a prime example of a complicated nonlinear multiphysics model, see [49]. Based on the principles of conservation of mass, energy, and momentum, the model consists of three PDEs that deliver the solutions for the fluid pressure p, the displacement of the porous rocks $\mathbf{u}$, and the temperature T. More precisely, one has to solve the following equation,

$$S\frac{\partial p}{\partial t} + \beta\frac{\partial T}{\partial t} + \nabla\cdot\mathbf{v}_D + \alpha\nabla\cdot\mathbf{v}_s = F \tag{40.9}$$

to obtain p. Here, S is the storage coefficient of the matrix, α is the Biot coefficient, and F is an external source, and the velocity of the solid $\mathbf{v}_s = \partial\mathbf{u}/\partial t$ is coupled to the displacement $\mathbf{u}$. Moreover, $\mathbf{v}_D = -\Lambda(\nabla p - \rho_f\mathbf{g})$ is the Darcy velocity and $\beta = (1-\phi)\beta_s + \phi\beta_f$, where ϕ is the porosity; β_s and β_f are the thermal expansion coefficients for solid and fluid, respectively; Λ is the flow mobility; ρ_f is the fluid density; and $\mathbf{g}$ is the acceleration due to gravity.

The temperature T is found from:

$$C\frac{\partial T}{\partial t} + C_f\mathbf{v}_D\cdot\nabla T - \nabla\cdot(\mathbf{K}\nabla T) = Q, \tag{40.10}$$

where C and C_f are bulk and fluid heat capacities, respectively; Q is a heat source; and $\mathbf{K}$ is the thermal conductivity tensor.

Finally, we have:

$$\nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{g} = 0, \tag{40.11}$$

where λ and μ are the Lamé constants, and $\epsilon = (\nabla\mathbf{u} + (\nabla\mathbf{u})^T)/2$, and

$$\boldsymbol{\sigma} = (\lambda\nabla\cdot\mathbf{u} - \alpha p - \beta_s(3\lambda + 2\mu)(T - T_0))\mathbf{I} + 2\mu\epsilon$$

is the total stress tensor as described by Hooke's law with a linear thermoelastic term. Here, T_0 denotes the initial temperature corresponding to vanishing thermal stresses.

The described model is relevant to our application only when it is applied for relatively short time intervals during which we expect relatively small deformations. That is, we are in a local regime where an elastic model is sufficient, although the larger-scale deformation of sediment packages will be mostly plastic.

Parallel Solution of Large-scale Basin Models

The coupled system built from equations (40.9)–(40.11) has five degrees of freedom for each grid node. Large-scale basin simulations require several million grid nodes, and therefore a parallel implementation of the solver is needed to solve the problem efficiently, see [42, 49]. In fact, the problem size can easily become too large for the discretised problem to be representable on a single processor. The parallel code is based on a finite element implementation in Diffpack, and it utilises the algebraic multigrid (AMG) method implemented in the ML software library [44] as a preconditioner. The discretised system is expressed on a block form, and the AMG sweeps implement the action of the blocks in a block-diagonal preconditioner. This preconditioner is combined with the Diffpack implementation of BiCGStab [65]. For an overview of parallel AMG methods, see [61].

The parallel solver has been tested on a series of benchmark problems with analytical solutions, and it also has been applied to a three-dimensional Petromod model of the Vøring basin, see Berdal Haga et al. [49]. The Vøring model has 1.5 million grid nodes and 7.3 million degrees of freedom, representing a 96.8 km by 90.8 km irregularly shaped basin consisting of 40 geological layers and 16 lithologies, see figure 40.8. The grid consists of tetrahedra and is unstructured.

To investigate the scalability of the proposed solution method, two types of performance tests were conducted: (a) considering a regular cubic geometry, one keeps the number of grid nodes per processor fixed to $32 \times 32 \times 32$; and (b) the Vøring model provides a fixed global problem size that is divided between the processors, resulting in fewer grid nodes per processor as more units are added. The numerical tests were run on a Cray XT4 computer with 1388 quad core processors. For the Vøring case, the software package ParMETIS [63] was used to partition the grid into smaller grid patches before distribution to individual processors.

For test case (a), efficiency measures of 71 per cent, 60 per cent, and 46 per cent were achieved for 512, 1024, and 2048 processors, respectively. That is, the time it took to solve the problem on 512 processors was roughly 360 times faster than solving it on one processor. Likewise, the speed-up was around 470 and 820 for 1024 and 2048 processors, respectively. The efficiency as a function of the number of processors flattened out as the processor count increased, which was expected due to the increasing collective communication cost.

For case (b), the efficiency measures were 56 per cent and 31 per cent for the runs with 512 and 1024 processors, respectively. These numbers correspond to speed-up values of 286 and 317. Clearly, the gain from adding more processors is very limited above a certain number. The lack of scalability for large processor counts occurs when the local problem size per processor becomes so small that the communication cost dominates the computational cost per local grid patch. A complete discussion of these and other experiments can be found in [49]. These results on scalability are comparable with recent studies of AMG methods on massively parallel computers [64], although the latter results are largely based on simpler geometries and grid structures.

40.6 Event-driven Construction of Geological Models

Numerical simulation of geological processes requires accurate representation of the geometry and the material properties of the involved rock layers. Traditionally, con-

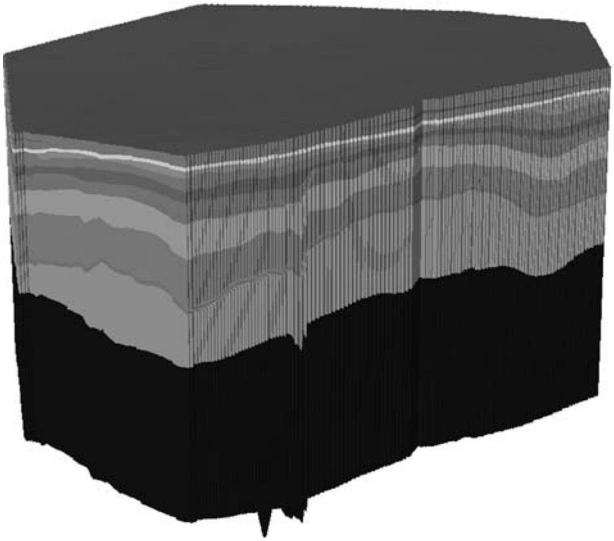


Figure 40.8 The Petromod model for the Vøring basin consists of 40 layers with 16 different lithologies. The geometry is embedded in a cubic domain covering 96.8 km × 90.8 km × 34.1 km with a horizontal resolution of 400 meters. The computational problem is defined on a grid with 1.5 million nodes, leading to 7.3 million unknowns. The Vøring data set is courtesy of StatoilHydro. (For the colour version, see figure C.16 on page 652.)

struction of geological models from seismic sections and other data sources has been focused on approximation of the involved geometry. The mathematical and numerical methods used for this purpose often have been based on methods designed for computer-aided design and similar modelling processes. Although such methods work well for the smooth geometries of an automotive body or an aircraft wing, adhoc tailoring of the methods has been required to handle the irregular structures of geological layers[59]. Once the geometry is established, the model must be populated with material properties throughout the computational volume. This spatially and temporally varying information must be derived from sparse measurements of the geology that is observable today. The techniques used for this parameter estimation are often taken from the field of geostatistics [66].

The traditional approach to constructing geological models has some evident weaknesses. First, constructing the geometry is very focused on details, and errors or misinterpretations can significantly alter the results of the process simulation. Due to the algorithms used for modelling, even small corrections of the input will lead to time-consuming and expensive editing of complicated geometrical models. Often, this editing must be performed in an iterative fashion, adding even more complexity to the modelling phase. Second, the extrapolation of locally observed material parameters throughout the volumetric model must be done consistently and in a way that preserves the basic physical properties of the system [60].

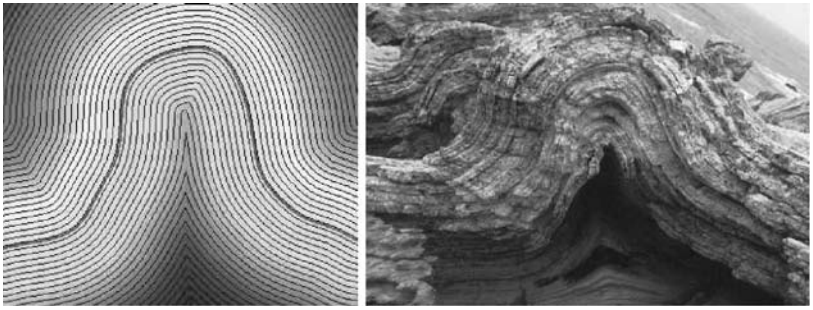


Figure 40.9 Left: A folded structure computed by the algorithms used by the compound model builder. Right: A folded structure observed in an outcrop. The illustrations are courtesy of Øyvind Hjelle.

Compound Modelling

Several research groups have tried to develop new paradigms for the construction of geological models that are easily editable and extensible [36, 67]. StatoilHydro has put considerable effort into the development of one such technology: *Compound Modelling* (CM). This technology originates from one of the company's senior researchers, Steen A. Petersen [37, 40, 39, 38]. The basic idea of CM is to view the model as a sequence of geological events placed along a geological timeline. The resulting model can represent geological structures of almost arbitrary complexity, including intricate relationships between horizons, faults, and physical installations such as wells. Figure 40.9 shows an example of a folded structure constructed by the algorithms used in the compound model builder, as well as a similar structure observed in an outcrop.

Technically, the CM approach leads to a tree structure of objects, in which each object describes a single geological event or represents the action of a submodel with its own tree structure. If one needs to edit this model, one can easily change individual events or the sequence of events. The actual geometry resulting from the model is implied by the computation of a *realisation* of a given configuration of the model. Similarly, the local observations put into the system will be used to consistently propagate the relevant material properties to all cells in the grid-based geometry. The combined realisation of the grid and the spatio-temporally varying geological parameters takes the form of a traditional geological model that can be funnelled into other applications, such as basin analysis or reservoir simulation. Put another way, the compound model can be seen as a program (tree structure) consisting of geological instructions (objects), the execution of which leads to a complicated result (realisation). Seen in this context, the shift from traditional modelling to compound modelling is similar to the shift from machine code to a high-level object-oriented programming language.

At the heart of the compound model builder, geological properties are distributed throughout the spatial domain based on the distances between each grid cell and the geometrical objects present in the model. Such geometrical objects can be points, curves, or surfaces of different types. Conceptually, the value of a geological property P_j in the point $\mathbf{x}$ is given as

$$P_j(\mathbf{x}) = C_{\text{geo-rules}}(p_j(\phi_i(\mathbf{x}), i = 1, 2, \ldots, N)). \tag{40.12}$$

Here, p_j is a one-dimensional property function that depends on distance, and ϕ_i is the distance between $\mathbf{x}$ and the ith geometrical object in the model. The functional $C_{\text{geo-rules}}$ represents an algorithmic representation of how the contributions originating from different geometrical objects are combined to form the property value P_j. This functional can be a nonlinear relation of high complexity. The spatial dimension enters the problem only through the computation of distances. Once the distance fields for each geometrical object are calculated, derivation of the property fields is related only to the one-dimensional concept of distance[11]. That is, the composition of property fields is independent of whether we are modelling in the two- or three-dimensional space. The clear division between rule-based composition of geological properties and geometrical distance fields provides a high level of flexibility that can be used in several ways.

Computing Distance Fields by the Fast Marching Method

Efficient and accurate methods to calculate distance fields is a critical component in the software used to construct compound models. In 2005 when the collaboration between Simula and StatoilHydro began, the compound model builder used a simple notion of axi-parallel distance, which can be used to model *similar deformations*. This application was limited to two-dimensional models. The technology was used primarily for quality assurance of the data interpretation, in that the synthetic seismic response computed by the model could be compared to the actual seismic image. Given a satisfactory match between the two, it is reasonable to believe that the interpretation present in the model catches the major features of the actual event. However, synthetic seismics represent just one type of realisation of a compound model, and a vast range of other uses are possible, both in exploration and production.

To date, our contribution to the CM technology has focused on advanced methods for calculation of distances and derived entities such as gradient fields, see Hjelle [46, 39]. To this end, we use the *fast marching method* [68], which computes the real distance based on the normal to the tangent at the curve or surface in question. This distance measure allows modelling of *parallel deformations*[12]. Because both similar and concentric deformations are geologically relevant, both types of distance measures coexist in the current version of the CM software.

The fast marching method can be generalised from two to three spatial dimensions. In fact, this methodology turned out to be the key to bringing the concept

[11] For some purposes, higher order derivatives of the distance fields are used, such as its gradient and curvature.

[12] In the literature, parallel deformations are also referred to as concentric. However, Ramsay and Hubert [69] state that this nomenclature is misleading and that the concentric deformation regime is a special case of parallel deformations.

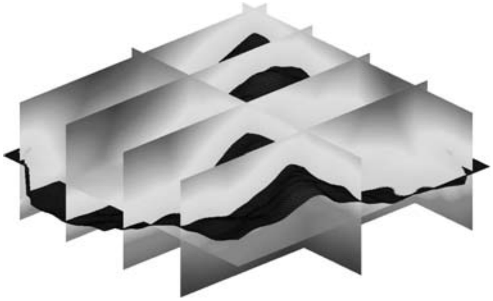


Figure 40.10 The three-dimensional distance field computed for a surface imported from an IRAP model. The illustration is courtesy of Øyvind Hjelle and Steen A. Petersen [46].

of CM from two to three spatial dimensions, see figure 40.10. By 2005, the original two-dimensional technology had been applied to about 65 cases, mainly for research purposes. The combined gain of handling three-dimensional problems and a broader range of deformation regimes has allowed StatoilHydro researchers to apply the CM technology to more field cases. Starting in 2006, the use of CM spread to production cases in Grane, Chinook, Troll, Oseberg South, and other fields. In total, CM has now been applied to more than 80 different cases.

The Eikonal equation. The fast marching method is used to solve the boundary value problem

$$\| \nabla T(\mathbf{x}) \| = \frac{1}{F(\mathbf{x})}. \tag{40.13}$$

The boundary condition is given as $T(\mathbf{x}) = 0$ for all $\mathbf{x}$ belonging to the geometrical object $\varGamma$, which is embedded in the computational domain. Here, T is the arrival time for the front starting at the boundary of $\varGamma$ and travelling at speed F. To pass an arbitrary point $\mathbf{x}$ only once, the speed function F is assumed to have a fixed sign. If F depends on position alone, the equation (40.13) is a classical nonlinear PDE referred to as the Eikonal equation[13]. This equation belongs to the family of stationary Hamilton-Jacobi equations. If we assume that the front propagates with a constant unit speed in the direction of the normal, we get $F = 1$ for all $\mathbf{x}$ and T coincides with the notion of the distance ϕ_i in (40.12). That is, we are then solving a problem of the form

$$\| \nabla \phi(\mathbf{x}) \| = 1. \tag{40.14}$$

Isocurves of the distance field, $\{\mathbf{x} \in \mathrm{I\!R}^d : \phi_i(\mathbf{x}) = \text{constant}\}$, are often referred to as level sets.

[13] In the Eikonal equation, F is required to depend on position alone. If F is allowed to depend on the time-varying shape of the propagating front, this condition fails. Such cases must be modelled with the general level set equation $T_t + F \| \nabla T \| = 0$ subject to the $T(\mathbf{x}, t = 0) = T_0$, which is an initial value problem.

The equation (40.14) can be discretised by finite differences defined on a Cartesian grid. An upwind scheme is required to generate an entropy-preserving solution that treats the cusps and corners correctly. In the context of the fast marching method, we use a modified version of Rouy and Tourin's numerical scheme [71] that provides second-order accuracy. The fast marching method is highly optimised and requires only $O(N \log N)$ arithmetic operations, where N is the number of grid points.

Derived Entities and Initialisation

In addition to the distance field ϕ_i resulting from the solution of (40.14) applied to the ith geometrical object Γ_i, the compound model builder would benefit from the computation of several derived entities. In particular, the gradient field $\nabla\phi_i$ and the curvature of ϕ_i are needed. The curvature field is closely related to the second derivatives of ϕ_i, and in the simplest case it reduces to its Laplacian, $\nabla^2\phi_i$. Therefore, the standard fast marching algorithm has been modified to compute these additional fields at the same time that it computes ϕ_i. The composition of geological properties in (40.12) also can benefit from the construction of *property distribution fields*. The value $P(\mathbf{x})$ of the property distribution field P is the property value measured at the point where the shortest path from $\mathbf{x}$ to Γ_i intersects Γ_i. That is, the field P is constructed by the orthogonality condition

$$\nabla P \cdot \nabla\phi_i = 0.$$

Put another way, the isocurves of P will be orthogonal to the isocurves of ϕ_i. The property distribution field P can be used to transport geological properties in target points at the geometrical object Γ_i consistently throughout the spatial domain. The modified formulation of the fast marching method takes all of these derived fields into account and calculates their values without changing the order of complexity of the algorithm. Figure 40.11 shows the relationships between a distance field ϕ, its gradient $\nabla\phi$, and an associated property distribution field P in the context of a simple geometrical object and parallel deformation.

Although the fast marching method has been applied to a wide range of problems, little information is available about how the method should be implemented to achieve the robustness and flexibility needed for a production-quality software tool. Experience shows that the method is very sensitive to the initialisation of the computations. By design, the fast marching algorithm computes the solution based on initial values in a narrow band around the given geometrical object. The method uses only one pass through the grid and therefore can not update or correct values that are inaccurate. Consequently, any noise present in the local solution originating from the narrow band will propagate through the domain and degrade the solution in other regions. The key to a robust and accurate solution is to provide high-quality data in the narrow band, which can be achieved by locally obeying certain smoothness criteria. The implementation of the fast marching method for CM uses spline curves and surfaces, high precision calculations of the closest point, and second-order finite differences to achieve smooth scalar fields, see [46]. The method can be extended

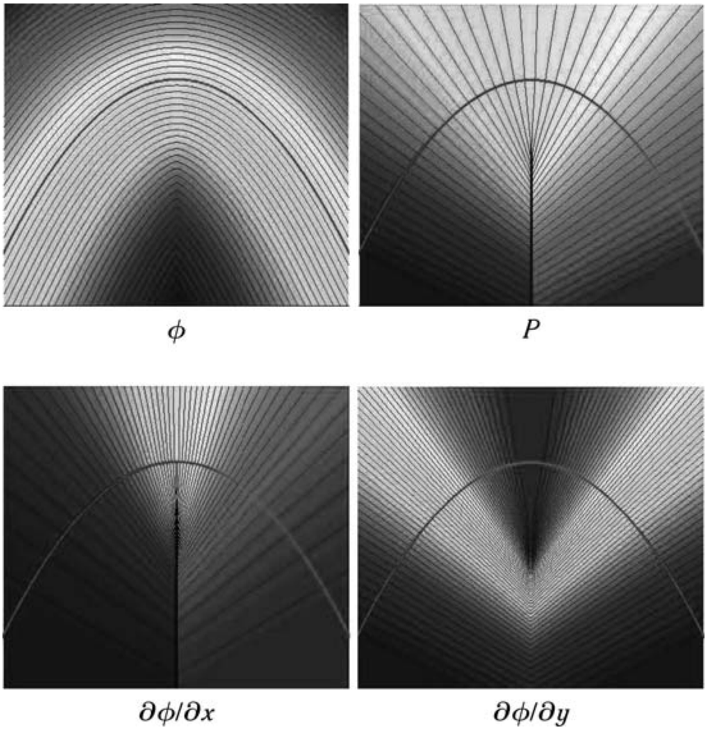


Figure 40.11 Top, left to right: The computed distance field ϕ and the parameter distribution field P. Bottom, left to right: First and second components of the gradient field $\nabla\phi$ ($\partial\phi/\partial x$ and $\partial\phi/\partial y$). The fields are associated with a simple parabolic curve and are presented in the context of parallel deformation. The illustration is courtesy of Øyvind Hjelle. (For the colour version, see figure C.17 on page 653.)

to higher order difference approximations if necessary. So far, the robustness of the developed method has been verified experimentally, and a more rigorous analysis is planned.

Non-trivial Deformation Regimes

The approach to computing distance fields presented herein has supported a robust implementation of CM that allows the representation of similar and parallel deformations. However, a much larger variety of geological deformations exist in nature. Several classifications of such deformations exist, but the one presented by Ramsay and Hubert [69] divides deformations into five classes, see figure 40.12. Recently, the fast marching algorithm in the compound model builder was extended to handle a generalised case of (40.13), in which the positive speed function $F(\mathbf{x})$ is allowed to vary with the orientation of the front. In particular, $F = F_{\text{prop}} + F_{\text{adv}}$ is taken to be a linear combination of a propagation speed and an advection speed. Here, the advection speed is:

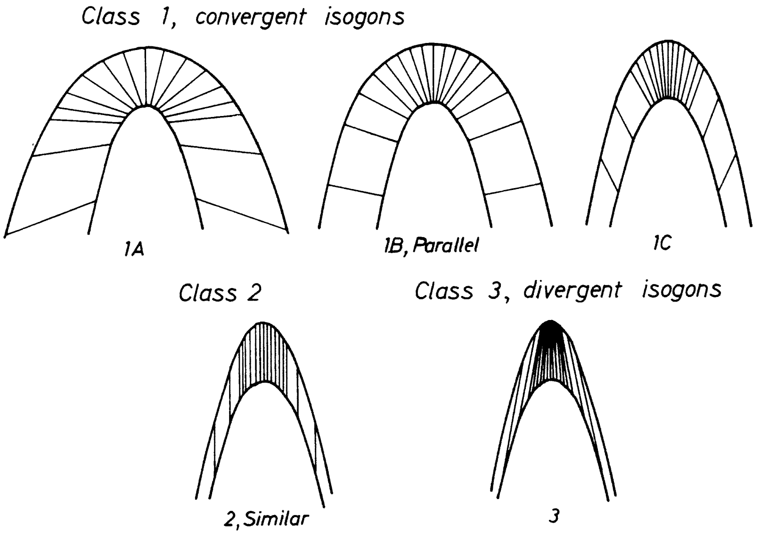


Figure 40.12 The deformation regimes classified by Ramsay and Hubert [69].

$$F_{\text{adv}} = \mathbf{U}(\mathbf{x}) \cdot \frac{\nabla T}{\| \nabla T \|},$$

where **U** is a given velocity field. This definition of F leads to the stationary Hamilton-Jacobi equation:

$$F_{\text{prop}} \| \nabla T \| + \mathbf{U}(\mathbf{x}) \cdot \nabla T = 1, \tag{40.15}$$

see [46]. Appropriate upwind discretisation of this equation makes it possible to compute the arrival time T. By varying the two speed components, F_{prop} and F_{adv}, all five deformation regimes classified in [69] can be recreate, see figure 40.13. Moreover, it is possible to span these classes of deformations continuously by varying the speed components. For further details on the modelling of non-trivial deformations, see [46].

To the best of our knowledge, the modelling capability introduced by the generalised equation (40.15) is novel. However, some of the features provided by the CM technology seem to be available through use of the (u, v, t)-transform, see [36, 59]. This methodology has emerged independently of the development of CM. Although the two modelling techniques share goals and some concepts, the overall approaches to the modelling differ.

Recently, the CM concept was extended further based on the observation that the property distribution present at any time can be viewed as the result of a recurrence of the process P acting repeatedly on the same rock volume. This observation led to a recursive model definition, referred to as *Earth Recursion*, see Petersen and Hjelle [40]. The current analysis of such recursions suggests that this approach can substantially improve the capability of shared earth model builders to express geological evolution. In particular, earth recursions are efficient at handling natural aging,

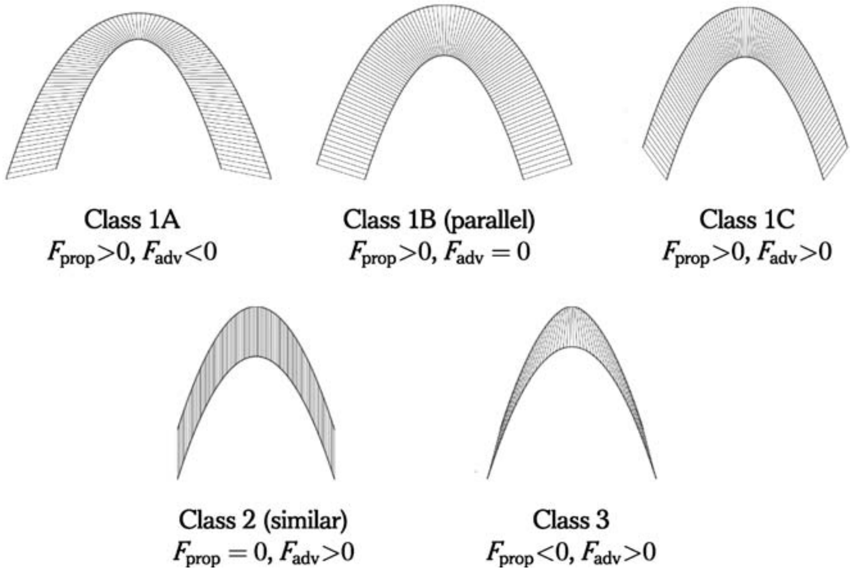

Class 1A
$F_{\text{prop}}>0, F_{\text{adv}}<0$
Class 1B (parallel)
$F_{\text{prop}}>0, F_{\text{adv}}=0$
Class 1C
$F_{\text{prop}}>0, F_{\text{adv}}>0$
Class 2 (similar)
$F_{\text{prop}}=0, F_{\text{adv}}>0$
Class 3
$F_{\text{prop}}<0, F_{\text{adv}}>0$

Figure 40.13 Deformations computed by different choices of F_{prop} and F_{adv} in the generalised Eikonal equation (40.15). The combinations of speed components shown here recreate the deformation classes of Ramsay and Hubert [69]. The illustrations are courtesy of Øyvind Hjelle and Steen A. Petersen [46].

property depth trends, incorporation of externally defined property distributions, and faults within volumes and structural restoration.

40.7 Visual Interaction with Multiscale Data Sets

The 2006 release of Google Earth[14] spawned enormous interest, both from professional users and the public. In its standard configuration, this virtual globe features a streamed three-dimensional display of satellite imagery of the Earth's surface, annotated with geographical information such as country borders, road networks, points of interest, and data related to weather and climate.

Imagine an application like Google Earth stripped of all touristy information and with support not only for images, but for general grid-based scalar fields of global or regional coverage; with advanced multiresolution rendering of multiple geological layers going subsurface; with rotation and dynamic masking of spatial information that vary with geological time; and, with highly optimised use of the computer's graphics processor for state-of-the-art visualisation and formula-based computation of new data sets. In short, these features are the main characteristics of the *4D Litho-*

[14] See earth.google.com.

sphere Model (4DLM) that Simula has been developing for StatoilHydro since early 2005, see Stensby et al. [45, 4].

Detailed analysis of the Earth's history requires us to combine vast amounts of different types of data that span several magnitudes of scale and spatial resolution. This is particularly true when one is searching for insight into complicated geodynamic processes. To fully understand the geological evolution of a region, information about the topography and the structure of the crust, lithosphere, and mantle must be combined with models describing the kinematics of tectonic plates. Such an integrated model can be a valuable tool for quantitative investigation of different tectonic phenomena, such as basin subsidence and heat flow history in extensional basins. Other parameters, such as stress distribution, changing vertical motion, and the corresponding changes of the sourcing and transport of sediments, are equally applicable for systems that are influenced by *plumes*[15] or tectonic compression.

The conceptual idea of 4DLM came from one of StatoilHydro's senior geologists, Jakob Skogseid. When the research collaboration between Simula and StatoilHydro began, he presented his vision for a software tool that would combine geological and geophysical data from different sources and of different scales in a novel way. Such a tool would fill a gap in the analysis needed for oil and gas exploration. The research and development of 4DLM has since been the flagship application generated by the collaboration between the two companies. Both CM and the 4DLM application are prime examples of a result that would not have been realised without the marriage of expertise from Simula and StatoilHydro.

The Basic Concepts of the 4DLM Implementation

In early 2005, an existing description of adaptive hierarchies of textures [23] served as the starting point for the implementation of the 4DLM. This description proposed a tile-based data structure defined on a planar domain that allowed the resulting surfaces to be textured, e.g., by images of a terrain. However, the concept of 4DLM required a spherical data representation. Therefore, we extended the technique in [23] by replacing the planar domain with a regular icosahedron[16]. This platonic solid can be seen as a polyhedral approximation of the sphere using 20 equilateral triangles as faces. From this geometrical base, more accurate approximations of the sphere are obtainable by successive refinement of the faces.

Similar to the original algorithm in [23], the 4DLM implementation uses the 4-8 grid hierarchy. The geometry is represented by a structure of diamond-shaped tiles, each of which comprises a regular grid with a fixed number of grid nodes in each direction. Each tile is linked to one or more parent (larger) and child (smaller) tiles.

[15] A mantle plume is an upwelling of abnormally hot rock from the mantle. The islands of Iceland and Hawaii are assumed to be the results of plume activity.

[16] In parallel with our extension to three dimensions, Hwa et al. [24] published their own three-dimensional version of the algorithm in [23]. However, their development was based on a cubic base. It is widely accepted that the icosahedral base, as used in 4DLM, is preferable. As the platonic solid with the smallest face size, the icosahedron leads to the smallest distortion in the transformation between the face and the corresponding spherical surface, see [25].

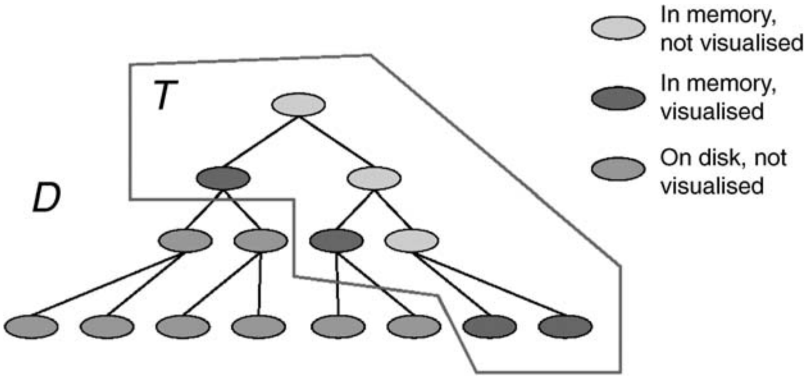


Figure 40.14 When a data set is loaded into memory, only parts of the associated persistent tree structure D (all nodes, regardless of colour) will be present. This subtree is denoted T (grey and green nodes), and only the nodes enabled for visualisation (green nodes) will contribute to the rendering of the scene. Be aware that the grey nodes conceptually will be in memory, even though all or parts of the data values are not yet loaded. (For the colour version, see figure C.18 on page 654.)

Given the fixed grid size per tile, the parents represent coarser information and the children account for finer details. Between each generation, the tiles are rotated 45 degrees. Traversing to a parent tile, the information for the new tile is the result of a low-pass filtering, while the traversal to a child tile calls for data interpolation. This hierarchy can roughly be thought of as a tree structure D, where each node in the tree holds the relevant data for a specific tile. For the purpose of the 4DLM, each node also carries a geo-reference that provides a link between the data representation and the global cartographic coordinate system. This geo-reference can be a pair of longitude and latitude values, a UTM zone, a mercator, or a similar global locator.

Given a viewpoint in three-dimensional space, the 4DLM software will choose only a subset of the tiles for the scene rendering, see figure 40.14. This subset T is chosen to be the smallest number of tiles that are sufficient to visualise the area of interest without introducing graphical anomalies. Consequently, tiles with a higher resolution are selected for the parts of the scene that are close to the camera rather than for the areas that are more distant. When the viewpoint changes, typically as a response to the user's interactive navigation in the scene, the subset T also changes. When updating the scene, tiles with a higher resolution will be introduced close to the new camera position, and tiles with a coarser resolution may be introduced in areas that have moved away from the camera.

Like other software systems for adaptive level-of-detail rendering, the 4DLM requires the data sets to be converted from standard formats to the tile-based representation outline above. This conversion is usually performed in a separate preprocessing step prior to the visualisation.

Further technical details concerning the implementation of the 4DLM can be found in the patent application [47].

Multiple Resolutions and Multiple Scales

The data sets used in a 4DLM session typically consist of grid-based information such as satellite images (pixel values) or surfaces representing geological horizons (depth values). The surfaces are originally constructed by other software systems, such as tools for seismic interpretation and geological model builders, and later imported into the 4DLM. By using the GDAL library[17], the 4DLM can read most industry-standard data formats, which are then preprocessed to construct the tile-based data structure required for an efficient level-of-detail rendering. The data sets can later be exported in a variety of standard file formats.

Each grid-based data set in the 4DLM is constructed from two basic items: the *geometry* of the surface and the *overlay* used for its colouring. Together, a geometry and an overlay constitute a *layer*. The collection of one or several layers, possibly extended with additional information such as point markers, line segments, polygons, or other types of annotation objects, is referred to as a *project*. While the geometry is based on the (x,y,z) coordinates for the surface, the overlay may be any relevant scalar field defined for the same (x,y) tuples. For instance, the overlay may be based on the depth (the z coordinate) or on physical properties such as porosity, pressure, facies, or magnetic anomalies.

In contrast to virtual globes such as Google Earth, the 4DLM can simultaneously visualise both global and regional data sets, possibly organised as a stack of subsurface layers, see figure 40.15. Moreover, the data sets loaded into a 4DLM project can be of different resolution. This opens up the possibility of using a finer resolution to capture the details of a small region, say an oil field, and still assess this information in the context of coarser, larger-scale data sets. It also is possible to combine several data sets with different resolutions into a new single data set; this occurs by seamlessly sewing the smaller data set into the larger one, see figure 40.16.

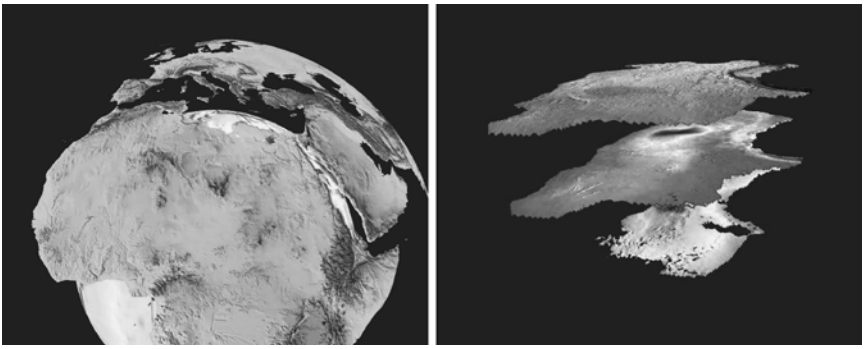


Figure 40.15 Left: This two-layer model consists of topography data with the water-covered areas cut away combined with a global surface indicating the bottom of the lower crust. Right: This stack of horizons from the North Sea has only local coverage. The illustrations are courtesy of Trond Vidar Stensby et al. [45]. (For the colour version, see figure C.19 on page 654.)

[17] See www.gdal.org.

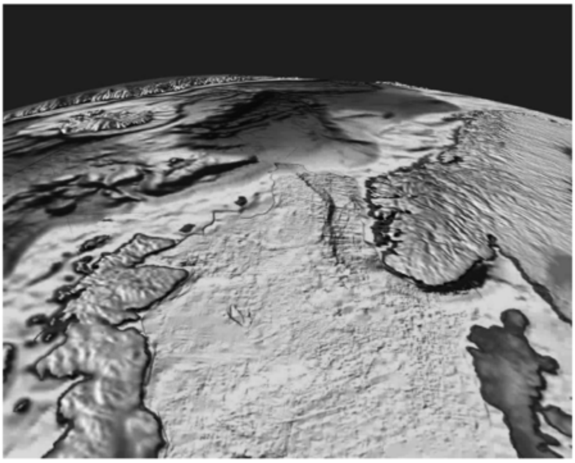


Figure 40.16 A high-resolution surface covering a small region in the North Sea has been sewn into a data set for the global topography. The illustration is courtesy of Trond Vidar Stensby et al. [45]. (For the colour version, see figure C.20 on page 655.)

Time-dependent Information

As pointed out in section 40.2, tectonic plates change their positions with time. For a detailed geological analysis, these changes need to be incorporated into the computational models. Therefore, the inclusion of geological time as the fourth dimension was a design goal for the 4DLM. However, several expert applications already exist for the creation and maintenance of *rotation models* that describe the kinematics of tectonic plates, such as Gplates[18] and Splates[19]. These models share some basic concepts. First, each tectonic plate is assigned a unique identifier, such as 701 for Africa and 201 for South America. Depending on the accuracy of the rotation model, up to several hundred plates of varying size may be involved in the model. Second, the movement of a plate relative to another plate is recorded as a sequence of *finite rotations*, each consisting of an Euler pole and a rotation angle, see [21]. A complete plate model contains a hierarchy of such rotations that describes the relative position of different plates at different points in time. This hierarchy can be represented as a time-dependent graph. The dependency on time reflects the fact that most plates have a distinct time of appearance and disappearance.

The 4DLM software can read standard rotation models described by a hierarchy of finite rotations. This information then can be assembled to rotation matrices that can be applied to the grid-based data in the current 4DLM project. For this process to work, the data sets loaded into the 4DLM must have been cut according to polygons that describe the outlines of the tectonic plates, and each resulting data fragment must have received its corresponding plate identifier. When the user selects a new

[18] See www.gplates.org.

[19] See www.geodynamics.no.

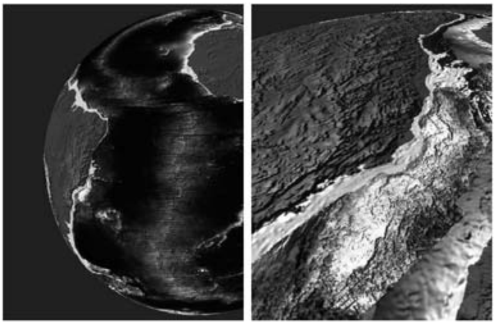


Figure 40.17 In the 4DLM, grid-based data can be moved with geological time according to an underlying rotation model. Here, this feature is illustrated by the positions of the tectonic plates for Africa and South America, including regional data sets for base salt layers outside Angola and Brazil. The images show the spatial positions today (left) and 114 million years ago (right). The illustrations are courtesy of Trond Vidar Stensby et al. [45]. (For the colour version, see figure C.21 on page 655.)

geological time point for the 4DLM project, the data fragments for each tectonic plate involved is rotated according to the loaded rotation model. In addition, the data fragments can be completely or partially masked according to their individual times of appearance and disappearance. In addition, the geo-reference present for each node in the data tree *D* must depend on time.

The rotation of spatial data according to geological time can be performed for both grid-based information and for vector data, such as isochrons[20]. For instance, figure 40.17 shows the relative positions of Africa and South America today and 114 million years ago. Similarly, figure 40.18 shows isochrons combined with grid-based topography in the South Atlantic today and 60 million years ago.

It is also possible to compute and visualise flow lines describing the relative motion between pairs of plates. If the user selects a point on a mid-oceanic ridge, the rotation model can be used for a given geological time to determine the points on the two adjoining plates that originated from the selected ridge point. Repeating this calculation for a sequence of time steps, one gets the path of a flow line as shown in figure 40.19.

To achieve the speed needed for interactive rotation of the grid-based data sets, both the spatial rotation of objects in the scene and the time-dependent masking of data are conducted by the graphics processor (GPU). The corresponding algorithms are implemented as special code fragments (shaders) that can be loaded into the GPU, see [62]. The processing power of the GPU also is used to implement fine-grained and completely interactive controls of colour maps, lighting, and surface scaling. Precise tuning of these visualisation parameters can reveal details in the geological surfaces that are not visible in a standard rendering of the same data sets.

[20] An isochron is a curve going through points of constant age on a horizon.

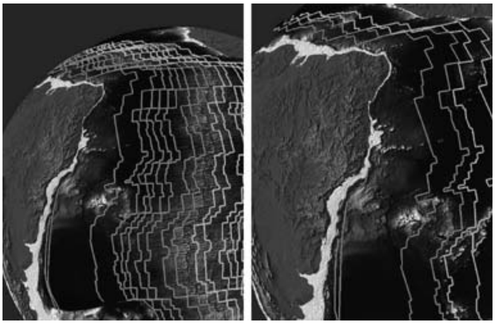


Figure 40.18 The 4DLM can rotate both grid-based and vector data, provided that appropriate plate identifiers have been assigned. These images show the locations of isochrons in the South Atlantic today (left) and 60 million years ago (right). The illustrations are courtesy of Trond Vidar Stensby et al. [45]. (For the colour version, see figure C.22 on page 656.)

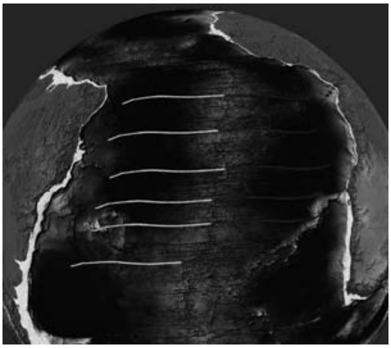


Figure 40.19 The flow line shows the time-dependent track of points that originated from a user-selected point at the mid-oceanic ridge in the South Atlantic. This illustration is courtesy of Trond Vidar Stensby et al. [45]. (For the colour version, see figure C.23 on page 656.)

Formula-based Computations

A main feature of the 4DLM is to represent and visualise data sets that contain a scalar value in each grid point (x, y). In its simplest form and by setting the scalar value to the depth (z value) in each point, such a data set can be the surface itself. In the general case, the scalar value can refer to any entity of geological relevance measured in the grid point, such as sediment thickness, porosity, or pressure. Given a collection of such scalar fields, it is technically possible to perform algebraic calculations for each grid point that involves the field values. Collecting the computed results of such algebraic manipulations provides one or more new scalar fields defined over the same grid points.

Several first principle approximations of geological processes can be expressed as formulas, and the 4DLM software implements a geological calculator that allows the user to define such formulas. For each variable in the formula expression, one can assign constant parameter values or attach one of the scalar fields already present in the model. Thus, it is quite easy to compute a large variety of relevant information that is not explicitly provided in the underlying data sets. The simplest example would be to compute the thickness of a geological layer using the formula $d = a - b$. By assigning the depth values of the top and bottom horizons defining the layer to a and b, respectively, the new field d will contain the thickness value in each grid point. More complicated formulas can deliver estimates of entities such as stretching factors, isostatic correction, paleo-heat flow, or paleo-water depth. For a slightly more complex example, assume that the current 4DLM project already contains data sets for the bathymetry h (field) and the free air gravity anomaly F (field). By defining the formula

$$B = F - 2\pi g \rho h,$$

where ρ is a selected density contrast (constant) and g is the acceleration of gravity (constant), the user can compute the Bouguer anomaly B. Formulas can also use geological time as a variable, thereby producing a time-dependent result.

The result of a formula-based computation in the 4DLM is a new scalar field that can be combined with geometry to define a new layer in the project. Depending on the chosen settings, the new field can be defined only by the presence of its formula, such that the resulting values are computed on the fly. Alternatively, it can be explicitly present as a collection of precomputed floating-point numbers. The resulting fields and associated layers can be visualised and manipulated, just like any other data set in the project, and they can be saved to disk for use in other 4DLM projects.

The computations needed to evaluate a formula are quite straightforward if all fields involved as operands are defined on exactly the same grid. If the fields have different spatial coverage or are defined on non-matching grids, special precautions must be taken. The computation results will then be defined in the spatial intersection of the involved domains, and interpolation between grids might occur.

Similar to the computations involved in the rotation of tectonic plates, the formulas are evaluated on the GPU. Each formula is automatically translated into appropriate shaders, which are executed for each relevant grid point. This approach ensures a high computational speed and permits interactive construction of the derived entities.

40.8 Former Research Activities

In the previous sections we outlined the ongoing contributions of the scientific collaboration between Simula and StatoilHydro. Here we describe some initiatives that are currently inactive.

The final report from the pilot project that started the collaboration included a discussion of the integration of the different steps in the workflows of oil and gas exploration. In reply to this discussion, Simula conducted research on *scientific workflows* that led to the pilot implementation of the Integrated Geological Model (IGM) software environment. This environment can create, maintain, and execute computing-intensive workflows wherein each work step can be tied to a specific software application. The primary goal is to improve integration among analyses conducted with different specialist tools, regardless of whether the application is an off-the-shelf commercial product or a home-grown research code. Moreover, the IGM can track the history of data generated in the different work steps, much like version control systems such as subversion[21] track source codes. When the user decides to roll the system back to a previously recorded state, the IGM will automatically reload the appropriate versions of input and output data for the different work steps in the workflow, thereby automatically taking care of data dependencies. The intent of this approach is to secure the quality of decision making by controlling the consistency and validity of data throughout the workflow.

Although the concept and architecture of the IGM was proven through its pilot implementation, its development has been discontinued. As predicted, the main obstacle preventing the success of this technology is the need to make the user's behaviour more uniform. To keep track of the generated data and their dependencies without having low-level source code access to the applications in all work steps, the IGM has to require a certain level of uniformity in the usage patterns. This requirement is not directly compatible with the established routines for typical exploration projects.

Simula also has been involved in studying how to model geological layers using measurements of the positions and dips of observable points in outcrops combined with various types of geological constraints. In short, this is a constrained optimisation problem, for which a solution procedure was devised based on an existing method for multilevel least squares approximation of scattered data over binary triangulations, see Hjelle and Dæhlen [11]. Currently, research on this particular application has been discontinued, but the developed concepts are being further refined through the CM project and other activities.

40.9 Key Factors for Successful Collaboration

Throughout the four years of collaboration between Simula and StatoilHydro, our research activities have covered many topics. Moreover, these topic have come from wide areas of research and application. Together we have managed to generate strong, and in some cases quite remarkable, results. In particular, CM and the 4DLM are both technologies of strategic importance to StatoilHydro. These technologies, and Simula's contributions to their creation and development, are described in sections 40.6 and 40.7, respectively. Moreover, ongoing research on calibration and quality assurance of computational results from depositional models is promising. It is still too early to tell whether the techniques described in section 40.4 will have industrial impact, but the potential of these two approaches seems significant and will be mapped out over the next several years.

Several observations about how the research collaboration in computational geosciences has been run stand out as noteworthy. First, the investment from StatoilHydro's side has been substantial, both in terms of funding and scientific commitment. Moreover, the investment had medium- to long-term goals from the very beginning. Over time, mutual confidence has been built, and now we see an even stronger de-

[21] See subversion.tigris.org.

gree of longevity in the plans for future collaboration. Consequently, we have been able to build up a sizeable research group that focuses on complicated problems that will require several years of work to accumulate results. Two other key factors for success were the selection of StatoilHydro personnel to take part in collaborative activities and the choice of research topics. From the beginning, StatoilHydro selected scientifically strong project leaders with years of industrial experience. These individuals have taken personal ownership in the ongoing activities; acted as mentors for the Simula personnel by communicating geological and geophysical knowledge in an efficient and understandable way; and made sure that their team members in the company have taken proper notice of the possibilities made available through the collaboration. The focused research areas have been carefully chosen to take advantage of the competence already present at Simula. This selection process follows a collaborative format, whereby both parties are free to suggest new topics. It is a mutually recognised principle that the selected research tasks must be strongly rooted in the research interest of StatoilHydro, either by contributing to existing focus areas or charting new possibilities for future practices.

The Computational Geosciences team at Simula consists mainly of researchers and software developers with a background in applied mathematics, physics, scientific computing, and advanced software design. Following a steep learning curve, the key staff members have acquired relevant geological understanding on a need-to-know basis, and the team has been extended by the addition of a postdoctoral fellow with a background in geophysics. As a whole, StatoilHydro's investment also has been an investment in people; it has built a task force that will serve the shared research interests well over time.

Internally at Simula, the possibility of building a strong, long-term collaboration with a major industrial partner has been taken very seriously. When creating the original research team, the members were carefully selected for their long experience, their ability to assimilate complicated information about a new field, and their solution-oriented approach to scientific problems. These people are still the backbone of the team, which after four years has more than doubled in size. Moreover, the team has been supported by Simula's management in several ways. First, management has strongly advocated that industrial collaboration of this kind should be conducted by Simula, and this view also is firmly supported by the Norwegian government, which is the major owner of the laboratory. Second, management has accepted that it takes time to build a foothold in a new field of research, and the requirements for scientific production in terms of journal papers has been reduced for the build-up phase. We now are ready to show our results to the public, and several papers are currently in production. Publication is also strongly supported by StatoilHydro, and more papers with joint authorship will soon appear. Moreover, without sacrificing the goal of a long-term basic research profile, Simula has acknowledged StatoilHydro's needs for continuous delivery of results that are useful to the industry. This need called for the establishment of Kalkulo AS, a subsidiary of Simula that can take on development-dominated tasks and deliver software products based on research. Recently, we observed that the link between basic research and technology development is bi-directional; activities that originally were mostly development-oriented now lead to new PhD and postdoctoral research projects. The intimate connection between the axes of research and development is a particular strength of the StatoilHydro-Simula collaboration. It should be noted that this feature is a consequence of the flexibility offered by Simula's organisation.

Finally, Simula's management has accepted a larger degree of autonomy in the Computational Geosciences group than that found in most other research groups at Simula. This freedom has allowed the group to build a very strong team spirit, which has helped each individual to meet the tough challenge of entering a totally new and complicated research field.

In summary, Simula's special organisation has allowed, and even nurtured, a research collaboration that is unique in Norway. This collaboration was made possible by StatoilHydro, which has proved to be a strategic and long-term partner. Both companies have taken ownership in the problems to be solved, and unusually strong professional bonds have been established between individuals and groups in the two organisations. Simula currently is exploring the possibilities of research collaborations in other industrial segments. The experience from the StatoilHydro collaboration serves as a model for these new enterprises.

Acknowledgements

This work was funded by research grants from StatoilHydro and the former petroleum division of Hydro. I would like to thank several employees of StatoilHydro who have been part of this collaboration since the very beginning. In particular, I thank Bjørn Rasmussen for his vision and enthusiasm, which made the collaboration possible. I also thank Ingolf Søreide for his strong involvement in the Simula School of Research and Innovation and Leif Lømo for his commitment to developing the academic research program of which SSRI is a part. I am grateful for the initiative, courage, patience, and advice that we have received from previous and current project managers in StatoilHydro: Jakob Skogseid, Tore Løseth, Allison Kennedy Thurmond, and Steen Agerlin Petersen. I also thank Ole Martinsen and Christian Zwach for their confidence in us as program manager and project owner, respectively.

The results reported in this paper would not have been achievable without a strong and enthusiastic group of researchers. The Computational Geosciences group at Simula has proven to be a first-class combination of clever individuals and a dependable team. I would like to thank PhD students Omar al-Khayat (particle-based methods for simulation of turbidity currents) and Joachim Berdal Haga (simulation of deformation and heat flow in sedimentary basins), as well as our postdoctoral fellow Stuart Clark (stochastic depositional modelling) and our researcher Hans Joachim Schroll (calibration of depositional models). I would also like to thank Øyvind Hjelle (CM and outcrop trace approximations), Arve Knudsen (workflow management), Trond Vidar Stensby and Christian Tarrou (4DLM), and all of the other employees of Kalkulo AS.

It is also timely to acknowledge researchers from other projects in the Scientific Computing department at Simula. In particular, our group has benefited from numerous discussions with and contributions from Hans Petter Langtangen, Harald Osnes, Xing Cai, and Bjørn Fredrik Nielsen. I am also very grateful for the strong and dedicated support from Simula's managing director, Aslak Tveito, in building up a brand new field of research inside the organisation. Finally, I would like to thank the former director of Simula, Morten Dæhlen, for his persistence in pursuing a research relationship with Hydro.

I would also like to thank the numerous persons from StatoilHydro and Simula who have provided valuable comments and feedback on this paper.

The work described in this paper was conducted at Simula Research Laboratory as part of CBC, a Centre of Excellence awarded by the Norwegian Research Council.

References

[1] M. A. Tatang, W. Pan, R. G. Prinn, and G. J. McRae. An efficient method for parametric uncertainty analysis of numerical geophysical models.*J. Geophysical Research*, 102:21925–21932, 1997.

[2] P. M. C. Leod, S. Carey, and R. S. J. Sparks. Behaviour of particle-laden flows into the ocean: Experimental simulation and geological implications. *Sedimentation*, 46:523–536, 1999.

[3] H. Li and D. Zhang. Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods. *Water Resources Research*, 43:523–536, 2007.

[4] A. K. Thurmond, J. Skogseid, C. Heine, T. V. Stensby, C. Tarrou, and A. M. Bruaset. Development of the 4D lithosphere model (4DLM): How exploration research has contributed to 4-dimensional visualization and interpretation of geological and geophysical data. *Eos Trans.*, 89:53, 2008.

[5] R. Walker. The origin and significance of the internal sedimentary structures of turbidites. *Proceedings of the Yorkshire Geological Society*, 35:1–32, 1965.

[6] J. Shirolkar, C. Coimbra, and M. Queroz McQuay. Fundemental aspects of modeling turbulent particle dispersion in dilute flows. *Progress In Energy Combustion Science*, 22:363–399, 1996.

[7] H. Li, H. Fang, Z. Lin, S. Xu, and S. Chen. Lattice Boltzmann simulation on particle suspensions in a two-dimensional symmetric stenotic artery. *Physical Review E*, 69(3):031919–+, Mar. 2004.

[8] Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover. GPU cluster for high performance computing. SuperComputing Conference, 2004.

[9] A. Masselot and B. Chopard. A Lattice Boltzmann model for particle transport and deposition. *Europhys. Lett*, 42:264, 1998.

[10] S. Huang, S. Mahadevan, and R. Rebba. Collocation-based stochastic finite element analysis for random field problems. *Probabilistic Engineering Mechanics*, 22:194–205, 2007.

[11] Ø. Hjelle and M. Dæhlen. Multilevel least squares approximation of scattered data over binary triangulations. *Computing and Visualization in Science*, 8(2):83–91, 2005.

[12] J. P. M. Syvitski and E. W. H. Hutton. 2D SEDFLUX 1.0C: an advanced process-response numerical model for the fill of marine sedimentary basins. *Computers & Geosciences*, 27(6):731–753, 2001.

[13] K. Borovkov. *Elements of Stochastic Modelling*. World Scientific Publishing, 2003.

[14] B. J. T. Morgan. *Applied Stochastic Modelling*. Chapmann & Hall, 2nd edition, 2008.

[15] B. D. Ripley. *Stochastic Simulation*. Wiley Series in Probability and Statistics. Wiley, 2006.

[16] W. Gautschi. Algorithm 726: ORTHPOL–a package of routines for generating orthogonal polynomials and Gauss-type quadrature rules. *ACM Trans. Math. Softw.*, 20(1):21–62, 1994.

[17] R. Ghanem and P. D. Spanos. Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, 57(1):197–202, 1990.

[18] L. Rainald. *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods*. Wiley, 2008.

[19] G. Hämmerlin and K.-H. Hoffmann. *Numerical Mathematics*. Springer, 1991.

[20] T. Løseth. *Submarine Massflow Sedimentation*. Number 82 in Lecture Notes in Earth Sciences. Springer, 1999.

[21] P. Keary and F. J. Vine. *Global tectonics*. Blackwell Publishing, 2nd edition, 1996.

[22] P. A. Allen and J. R. Allen. *Basin Analysis: Principles and Applications*. Blackwell Publishing, 2nd edition, 2005.

[23] L. M. Hwa, M. A. Duchaineau, and K. I. Joy. Adaptive 4-8 texture hierarchies. *Visualization Conference, IEEE*, 0:219–226, 2004.

[24] L. M. Hwa, M. A. Duchaineau, and K. I. Joy. Real-time optimal adaptation for planetary geometry and texture: 4-8 tile hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, pages 355–368, 2005.

[25] K. Sahr, D. White, and A. J. Kimerling. Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, 30(2):121–134, 2003.

[26] J. Rivenæs. Application of a dual–lithology, depth–dependent diffusion equation in stratigraphic simulation. *Basin Research*, 4:133–146, 1992.

[27] J. Rivenæs. *A Computer Simulation Model for Siliciclastic Basin Stratigraphy*. PhD thesis, NTH, 1993.

[28] B. Doligez, D. Granjeon, P. Joseph, R. Eschard, and H. Beucher. How can stratigraphic modeling help to constrain geoststistical reservoir simulations? *Numerical Experiments in Stratigraphy: Recent Advances in Stratigraphic and Sedimentologic Computer Simulations*, volume 62 of *SEPM Society for Sedimentary Geology Special Publications*, pages 239–244. Geological Society Publishing House, 1999.

[29] D. M. Tetzlaff and J. W. Harbaugh. *Simulating Clastic Sedimentation*. Van Nostrand Reinhold, 1989.

[30] L. Landweber. An iteration formula for Fredholm integral equations of the first kind. *Amer. J. Math.*, 73:615–624, 1951.

[31] H.-J. Schroll. Automatic calibration of depositional models. *Automated Solution of Differential Equations*. Springer, 2009. Submitted by invitation.

[32] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Book Series on Optimization. SIAM, 2009.

[33] D. Zhang. *Stochastic Methods for Flow in Porous Media. Coping with Uncertainties*. Academic Press, 2002.

[34] K. Bitzer and R. Salas. SIMSAFADIM: 3D simulation of stratigraphic architecture and facies distribution modeling of carbonate sediments. *Computers & Geosciences*, 28:1177–1192, 2002.

[35] J. Strobel, R. Cannon, C. Kendall, G. Biswas, and J. Bezdek. Interactive (SEDPAK) simulation of clastic and carbonate sediments in shelf to basin settings. *Computers and Geoscience*, 15:1279–1290, 1989.

[36] J.-L. Mallet. Space-time mathematical framework for sedimentary geology. *Mathematical Geology*, 36(1), 2004.

[37] S. A. Petersen. Compound Modelling - a geological approach to the construction of shared earth models. *EAGE 61th Conference & Exhibition, Extended Abstracts*, 1999.

[38] S. A. Petersen. Optimization strategy for shared earth modeling. *EAGE 66th Conference & Exhibition, Extended Abstracts*, 2004.

[39] S. A. Petersen, Ø. Hjelle, and S. L. Jensen. Earth modelling using distance fields derived by Fast Marching. *EAGE 69th Conference & exhibition, Extended Abstracts*, 2007.

[40] S. A. Petersen and Ø. Hjelle. Earth recursion, an important component in sheared earth model builders. *EAGE 70th Conference & exhibition, Extended Abstracts*, 2008.

[41] M. G. Imhof and A. K. Sharma. Seismostratigraphic inversion: Appraisal, ambiguity, and uncertainty. *Geophysics*, 72(4):Rr51–R66, 2007.

[42] J. B. Haga, A. M. Bruaset, X. Cai, H. P. Langtangen, H. Osnes, and J. Skogseid. Parallelisation and numerical performance of a 3D model for coupled deformation, fluid flow and heat transfer in sedimentary basins. *MekIT'07*, pages 151–162. Tapir Academic Press, 2007.

[43] O. Al-Khayat, A. M. Bruaset, and H. P. Langtangen. Lattice Boltzmann method and turbidity flow modeling. *MekIT'07*, pages 213–228. Tapir Academic Press, 2007.

[44] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 Smoothed Aggregation User's Guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[45] T. V. Stensby, C. Tarrou, A. M. Bruaset, J. Skogseid, A. K. Thurmond, and C. Heine. Multi-resolution visualization of time-dependent horizons on the globe. Presentation at the 33rd International Geological Congress, Oslo, 2008.

[46] Ø. Hjelle and S. Petersen. Mathematics of folding in structural geology. Working notes, 2009.

[47] StatoilHydro and Simula. Interactive rendering of physical entities. UK Patent Application No. 0814474.3, filed August 6, 2008, 2008.

[48] O. Al-Khayat, A. M. Bruaset, and H. P. Langtangen. A lumped particle model for the simulation of suspended flows. Journal paper in writing, 2009.

[49] J. B. Haga, H. Osnes, and H. P. Langtangen. Parallelisation and numerical performance of a large-scale porothermoelastic basin model. Journal paper in writing, 2009.

[50] S. Clark, A. M. Bruaset, T. Sømme, and T. Løseth. Probabilistic handling of uncertainty in diffusion-based stratigraphic models. Journal paper in writing, 2009.

[51] T. Salles, S. Lopez, R. Eschard, O. Lerat, T. Mulder, and M. C. Cacas. Turbidity current modelling on geological time scales. *Marine Geology*, 248:127–150, 2008.

[52] A. Kjeldstad, H. P. Langtangen, J. Skogseid, and K. Bjørlykke. Simulation of sedimentary basins. *Advanced Topics in Computational Partial Differential Equations*, pages 611–658. Springer, 2003.

[53] H. P. Langtangen. *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming*. Springer, 2nd edition, 2003.

[54] M. J. D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Math. Program., Ser. B*, 92:555–582, 2002.

[55] DK Images. http://www.dkimages.com/.

[56] Jerome Neufeld. Photo from a tank experiment showing a turbidity current.The experimental nonlinear physics group, University of Toronto, Canada. See http://www.physics.utoronto.ca/~nonlin/turbidity/turbidity.html.

[57] Kevin Walsh. Photo of a stack of turbidites in Cornwall, UK. See http://www.everystockphoto.com/photo.php?imageId=8672.

[58] A. B. Watts. *Isostasy and Flexure of the Lithosphere*. Cambridge University Press, 2001.

[59] J.-L. Mallet. *Numerical Earth Models*. EAGE, 2008.

[60] J.-L. Mallet. *Geomodeling*. Oxford University Press, 2002.

[61] U. M. Yang. Parallel algebraic multigrid methods – High performance preconditioners. *Numerical Solution of Partial Differential Equations on Parallel Computers*, pages 209–236. Springer, 2006.

[62] M. Rumpf and R. Strzodka. Graphics processor units: New prospects for parallel computing. *Numerical Solution of Partial Differential Equations on Parallel Computers*, pages 89–132. Springer, 2006.

[63] G. Karypis, K. Schloegel, and V. Kumar. Parmetis parallel graph partitioning and sparse matrix ordering library, version 3.1. Technical report, University of Minnesota, 2003.

[64] W. Joubert and J. Cullum. Scalable algebraic multigrid on 3500 processors. *Electronic Transactions on Numerical Analysis*, 23:105–226, 2006.

[65] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1992.

[66] R. A. Olea. *Geostatistics for Engineers and Earth Scientists*. Kluwer Academic Publishers, 1999.

[67] M. Perrin, B. Zhu, J.-F. Rainaud, and S. Schneider. Knowledge-driven applications for geological modeling. *Journal of Petroleum Science and Engineering,* 47(1):89–104, 2005.

[68] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge University Press, 1999.

[69] J. G. Ramsay and M. I. Hubert. *The Techniques of Modern Structural Geology: Folds and Fractures.* Academic Press, 1987.

[70] C. M. R. Fowler. *The Solid Earth: An Introduction to Global Geophysics.* Cambridge University Press, 2nd edition, 2004.

[71] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal,* 29(3):867–884, 1992.

41

A TALE OF THREE START-UPS

An interview with Christian Tarrou, Hans Gallis, and Viktor Eide by Dana Mackenzie

One of the characteristic features of the “Simula culture” is the strong encouragement given to researchers who want to become entrepreneurs. To find out how effective this support has been, we interviewed the chief executive officers of three new businesses that have spun off from Simula Research Laboratories. There were no holds barred in this interesting discussion of the trials and tribulations of founding a business.

Hans Gallis is the CEO of Symphonical, a company that has developed a web-based platform for building, sharing and running all kinds of processes (everything from creative processes to more structured software engineering processes). As he explains below, his company did not really grow out of a focused research effort at Simula; it could be described as a product of opportunity. His experience is also unusual because his company was the first of the three to launch. The role of trailblazer was both an advantage and a drawback, as you will see below.

Christian Tarrou is the CEO of Kalkulo, a subsidiary of Simula that provides consulting and development services to such Simula clients as StatoilHydro. His experience also differs from the other two managers in important ways. Kalkulo remains much more closely linked to Simula than the other two companies. To push the “spin-off” analogy perhaps farther than it should really go, Kalkulo is a satellite that still remains in orbit around the mother planet, while the other two companies have left Simula’s gravitational influence.

Viktor Eide is the CEO of Lividi, a company that sells streaming media software that automatically adapts to the user’s available bandwidth. Lividi (initially an acronym for Live Video Distribution) was the most recent of the three companies and was formally established as a company in 2008. By that time Simula Innovation had been operational for three years as a resource for new start-ups, while Gallis

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_41,

formed his company at essentially the same time that Simula Innovation was getting started.

"Viktor, could you tell us what your company does, and give us a brief account of how you founded the company?"

Viktor Eide (VE): "We provide solutions for streaming live media in an adaptive, or as we refer to it, an intelligent way. We see an increasing heterogeneity today both in devices and computers, from handhelds to stationary computers. We also see heterogeneity in networking technology, ranging from wireless (with lots of variability with respect to available bandwidth) to high-capacity wired networks. The technology we provide allows streaming systems to be deployed that are able to operate in this highly varying environment."

"Is Lividi an outgrowth of your PhD work?"

VE: "At least partly. I worked in a project on real-time video content analysis, which is useful in, for example, video surveillance applications. My interest was related to efficient distribution of sensor data, in particular video data, which is challenging due to the huge amounts of bandwidth. We researched how to efficiently distribute video and real time data, not only from one source to one recipient but one-to-many and many-to-many in an efficient and adaptive manner."

"At what point did it occur you that this could be a wide use, that you could do it on the Internet and you didn't have to limit it to just this video surveillance application?"

VE: "I think that was fairly clear all along, video surveillance was just an example application for our research."

"When did you decide that you were going to spin off your own company?"

VE: "In spring 2006, we contacted the technology transfer organization (TTO) at the University of Oslo and Simula Innovation. It was decided that Simula Innovation should take the main responsibility for Lividi, due to where people had been working most. After talking to these TTOs in 2006, we applied for a verification grant from the Research Council of Norway. We got this funding in spring 2007 and we employed two full-time employees in 2007. Now we have three full-time employees and some part time.

"I should point out that it was not only push, it was also pull. The TTOs come to you and ask if you have something that might be of commercial interest. Since around 2003 in Norway, the employment contracts now say that you have to inform them about results with commercial potential. The new rules changed another thing with respect to rights. Before that point in time, the researchers owned their own results. After that time the research institutions also have some of the rights."

"Could you talk about these verification grants? What do you have to do for them?"

VE: "When technologists go to TTOs, the grant is to verify that there is some commercial potential. We did two things. From one side, we worked on the business model and identified market segments and players in those market segments. We set up meetings with some of these. And then we developed technology demonstrations that we could bring with us out to the customers and show how we could potentially create value for them. We had this generic technology for doing adaptive streaming management. But just trying to sell that to a potential customer is not necessarily the right thing. You need a demonstrator that is able to show the benefits for that particular customer in their particular market segment. So the technology and the business must go hand in hand."

"Hans, can you talk a little bit about your company, what you do, and how that came out of your work in Simula?"

Hans Gallis (HG): "Yes, I started at Simula in June 2002. I worked on a PhD within process engineering (software engineering). At that time there was a lot of hype about agile software development processes. I started focusing on those processes, and how they could be helped by different types of software tools. I found that the communication module is probably the most important part of the technology, because that sort of iterative, incremental development process in short cycles demands lots of communication. Then I actually ended up by studying the smallest part of the collaboration, which is when two people develop software together. There was too much noise when I tried to study a bigger group! I compared how two people solve problems compared to how one individual solved the same problem. It was more psychology than it was informatics."

"How did you study this?"

HG: "My supervisor, Erik Arisholm, who is working here at Simula, got money from the Norwegian Research Council to run at that time the world's largest controlled experiment in software engineering. One hundred individuals and one hundred pairs were hired to solve Java tasks. We compared how they solved tasks in pairs and as individuals.

"Although this wasn't the main result of the research, we noticed, through case studies, that very few actually used huge project management tools to handle their daily project tasks. A lot of professionals ended up using Excel spreadsheets instead of specific developed tools for running the process. I analyzed more than 100 different project management tools, and they were very similar. In all the tools you had to call something a milestone, an activity, or a task. But what about ideas that popped up, what about all the other stuff that is happening around the project? Agile software development is focusing a lot on innovation and rapid change in plans and requires a very different approach to development processes than traditional so-called waterfall models, in which all the stages are planned in advance. The engineers found that the tool was actually an obstacle to their work, and they would go back to using Excel spreadsheets or email or Post-it notes or Word documents, because they are more flexible and give them more control and support better how they actually work. They don't have to be stopped from doing their work because of a tool.

"The reason for starting Symphonical was that we are competing not with 100 development tools, but (in our opinion) with only one *type* of tool. We saw this as a good opportunity to do something completely different, and in a way that is more flexible and not working as an obstacle. We chose the name Symphonical because we wanted to distinguish ourselves from the management perspective and the traditional notion of 'project management.' It came out of the idea of orchestrating instead of micromanaging. All of those other 100 tools have been developed from a management perspective; they were developed by managers for managers. We wanted to look at the interactive part of the development process."

"What's the business model? How do you plan to make money? The website all looked free."

HG: "That's a hard question. It has turned out that the most difficult part of a business is not the technology. Of course it might be difficult to solve certain technology problems. But it's turning out that the business model is much more difficult than I ever expected. I know that everybody tells you to work on the business model from day one, to focus on the business model and not the technology, because it's easy for a technologist to focus on the technology. Everyone has told me that from the beginning, and I still have to say that I have not focused enough on the business model!"

"Are there experiences that make you say now, 'I wish I had known that when I started?'"

HG: "Oh yes! One example was the business model, to focus more on the business aspect. Anther thing I would have done differently is to take more breaks. To stop developing, to look back, to have one week or maybe a whole month when you just lay down everything and ask yourself: 'Are we targeting what we agreed on, or are we going in another direction?' We have been sprinting a little bit too much, because this is a marathon."

"Victor, in your experience did you think you were sprinting too much, or did you plan on a marathon?"

VE: "You know the statistics: Most companies take a long time to get established. You have to be prepared to do it for quite some years."

HG: "I think that it's very similar to doing a PhD It's not done within a year; you have to work with it for three or four years and often many more years."

VE: "I think it's different in many ways. A PhD is, at the end of the day, your own work. You're the responsible person in the end. When you try to establish a company, it's more about teamwork."

Hans Gallis, Christian Tarrou, and Viktor Eide

"Do you have to change your thinking?"

VE: "Somewhat. When you're doing research, the product is publications. Success is if reviewers and your peers think that your work is interesting. In business it's about getting customers, and having someone to pay at the end of the day for your product. It's somewhat different but still there are lots of similarities. You have to be able to sell."

HG: "It's a different kind of selling. You don't have to sell your message. If you go talk to customers, they don't care if you can argue really well; they are more interested in the benefits of your product... You have to use a very different set of arguments. A lot of researchers can be good entrepreneurs, but some would be really bad. I don't think entrepreneurship is for everybody."

"Do you like being an entrepreneur better than being a graduate student?"

HG: "I think this is more right for me than doing an academic career. I was never about to do an academic career. I never finished my PhD, but if I had finished I would never have done a postdoc. I'm really eager to learn, and running my own business is part of it."

Christian Tarrou (CT): "I think one of the challenges for all three of us is that we are first of all, by formal training, technologists. Business development is another formal training, actually. We come from the other side. Both ways can lead to a good result, but that is one of our main challenges, to adopt that kind of business thinking.

"You asked if Simula Innovation contributes to that. That is the idea, that Simula Innovation shall contribute actively, both by itself and also by external contacts, to help forming new businesses. Of course, Simula Innovation is a young company, and they are themselves in the process of becoming a better TTO for Simula. Slowly we are getting to where we should be."

"Christian, when did you come here, and how did Kalkulo get started?"

CT: "I started in 2004, in the Scientific Computing department. I worked there for a couple years. Kalkulo was formed because of the StatoilHydro collaboration. I was not the manager of Kalkulo at that time. The former director left one and a half years ago and I took over.

"During the pilot project with StatoilHydro, it was realized that certain areas of common interest needed development services, not only research services. In order to split the research and development activities, Kalkulo was formed. In the beginning there was one director and four who worked on projects."

"At what point did you become the director?"

CT: "At the point where the former director and three programmers were leaving Kalkulo to form a separate company called World Beside. At the time, Kalkulo lived off the StatoilHydro collaboration and also had a project called World Beside. The activity supporting StatoilHydro remained in Kalkulo and the other was formed as a separate company. Kalkulo was split in half, basically. Four people quit and three remained."

"What were some of the biggest surprises or things you've had to learn as CEO?"

CT: “Surprises is perhaps not the word, but challenges or problems are numerous. Basically, I have had to learn lots of things, such as handling personnel, recruiting, managing projects, budgeting, selling new projects, and following the economy. And also the business plan aspect, becoming more aware of where we are going, what should be the business model.”

“Contrasting with Hans’ business, it seems that you have one major client at this point. Are you looking for more clients?”

CT: “It is correct that since World Beside was spun off, StatoilHydro has been the biggest client, the one big client. We have had other projects as well, but the bulk has been StatoilHydro. One important difference in the business model is that we deliver services. We don’t make a product yet. That may be in the future. Also, we have in some sense earned money from day one. We have not been in a situation where we needed money from external investors.”

“Viktor, who are your customers or clients?”

VE: “We are now involved in two innovation projects. Those are projects running, one for three years, the other one for four years with an optional extension of three years. They are pretty long duration, which gives us some base funding. At the same time they provide something else. We applied to those projects because we think we can develop our technology further, and we can establish relations not only in Norway but internationally. One of these projects is called March. It has a lot of European partners in addition to some Norwegian ones. These projects may also give us a lot of visibility. It wasn’t only about the money but also about these other things.”

“Hans, at what point did you declare your independence from Simula?”

HG: “Our company is not attached now, and was not ever attached to the research done here. Staying here was maybe a good idea for the other companies because you have your environment here, and you have access to the best researchers. We moved out of here pretty early because we actually wanted to get away from the research environment and focus more on the business. So we moved to the city centre, instead of staying out here.

“This is something that I think could have been improved. Fornebu was going to be this great Silicon Valley place, but it hasn’t worked out that way... ”

VE: “It’s too early to say.”

HG: “Yes, it’s too early to say. I agree. We need more of this... I don’t like the word incubator, because I’ve seen so many situations where the incubator doesn’t work at all. At the time that we spun out as a company, there weren’t that many other companies to discuss with, to share thoughts with, and I think that’s something we missed: a more formal structure that made it more obvious why we should stay here, and what were the benefits to staying here as part of the Simula environment. For us that was totally unclear, and it was better to be in the city centre, because it was a shorter distance to the customers. I think from my perspective that the Simula evaluation could do something about this... I think they should take charge of it. Not necessarily being a typical incubator, because I'm not sure if that works either, but somehow trying to at least grasp the benefits of being in the Simula environment. I think it's a great thing if researchers who are interested in being entrepreneurs can share their thoughts. Also, I think researchers and entrepreneurs could benefit from each other's knowledge and experience. In my opinion Simula Innovation should focus more on integrating the knowledge of the entrepreneurial researchers."

CT: "I guess it's pretty clear that it should be Simula Innovation's responsibility. To sum it up, what we would expect from Simula Innovation is, since we come from this technology world, to provide help with all of the aspects of development and innovation and entrepreneurship... Another part is that Simula Innovation should possess the necessary competence themselves or that they have preferred partners that we could use for everything from contracts to business plan development."

"Hans, what are some thoughts would you like to share with other entrepreneurs?"

HG: "What I have learned is that the most important thing we did was to involve an experienced board of directors. The money is not that important. You can develop things with half the money. But all the experience that our board has provided is worth more than the money they have provided. Also, I think a very important experience is to be patient and to 'hang in there'. Things take longer time than you expect. That doesn't mean that you are doing anything wrong. You should give yourself the time to err and learn. Most companies need at least five to ten years before they are really profitable. Only one in ten thousand companies is a Google or a Facebook."

CT: "Starting up a company is a very difficult and challenging journey, and we are doing it the first time. There are persons that you have on a board of directors who have probably done this journey lots of times, and that helps enormously."

VE: "I have to say I agree very much with Christian here. When technologists go to a TTO and the TTO takes half of the shares (or two-thirds of the shares at a university), then I think it's fair to expect a significant contribution to all the other areas besides technology. You have mentioned some of them, but there are lots of other areas where you need competence: contracts, economy, basic office functions, rules with respect to funding, where you can apply for grants. There is a long list of things that you need to learn to do.

"When we started, we got lots of support and enthusiasm from Simula Innovation. They got us in contact with people who helped us in business development. We got the right focus, not only focusing on the technology but also on the potential customers."

"Hans was saying that he thought the money was not as important as the expertise of the board. Do you disagree with that?"

VE: "If you don't have money, you have nothing. You need money. If you don't have that, you don't need the other things."

HG: "I agree that it's a basic need, but my point was that if you have a choice of 4 million or 2 million kroner, I'd swap the 2 million extra for a board with a lot of experience. I don't know if that makes sense, but that's how I see it."

VE: "To me the money is the resources. You need that for employing people. If you want to develop a product and you need to invest development time, you need money. Money gives you the opportunity to buy competence, to attract people. If you want really competent people on the board, you need to compensate them as well. Otherwise they have other things to do."

HG: "I think it's a chicken-and-egg problem, because if you have a really competent board, you'll definitely get money. No problem. Our board has a huge network, and that lets me relax and focus more on what I can do, the technology and that stuff. Instead of struggling to get 2 million kroner, they can get 10 million, more or less, just like that."

"Where did you get the members of the board from?"

HG: "It was based on hard work and luck, I guess. My sister has gone to the North Pole. At that time she was 21, and I think she was the youngest ever. There's kind of a North Pole club, and some of those people have quite a lot of money, because they have paid for the trip. She just found out that one of those guys, Bjørn Haugland, had founded several companies, and she knew that he was quite rich as well. She put us in contact with him, we presented to him and he liked the ideas. At the time he was looking for some new opportunities. It was kind of luck, but you should have some luck. We had meetings with Simula Innovation and with him, and he got in more people that he knew who were willing to be on the board and spend 500,000 or a million kroner, and suddenly we have a fair amount of business angels.

"When I look back at it now, I think that using business angels instead of venture capital companies at that early stage was a really, really great thing for us. We have direct access to the money, and we have direct access to the brains, instead of going through some young business guy who is watching over the money from the VC's, so that you never get to talk to the people that actually own the money.

"A lot of people talk about venture capital companies, but I think at an early stage there should be more structured business angel networks. VC's are usually something you go to when your product is more mature. They always come up with these term sheets. If they give you 2 million kroner, you have to reach those goals. If you don't reach that goal, they will convert more of their money into ownership and then you're out of business. At the beginning you should have time to make mistakes. Business angels know that you need time to make mistakes."

"Viktor, it seems to me that you have time because of the grants you got. Do you think the grants give you time to make mistakes?"

VE: "No. I don't think so. It gives you some flexibility, some opportunity to say no to some customers or some contracts that you think are not good enough for your company. I don't think it gives you the ability to make more mistakes. When you are starting a company, making mistakes–you should try to avoid that."

"Christian, do you have to worry about finding money and experience, or do you work with the structures that you already had, because Kalkulo had already started?"

CT: "Yes and no. We are in a different situation. We have some of the same challenges, and we have different challenges. Since we deliver services and since we started up having one client, we had money from the beginning. But we were very small to begin with and vulnerable economically. There is a bigger overhead when you are small. If that one project is cut down, then you are back to zero. My challenge is to get more projects, more consulting services projects from customers other than StatoilHydro. We have a pilot project for Statkraft. We are looking for more customers.

"One important point with Kalkulo is that we want to stay close to Simula for several reasons. The nature of the StatoilHydro collaboration is an example. We work well together. Large companies like StatoilHydro and possibly Statkraft are interested in these mixed research and development services. From my point of view, staying close to Simula is good for visibility and credibility. Kalkulo is very small (it's now nine persons), and Simula is ahead of us in terms of recognition. For us it's good to brand ourselves as part of the Simula Research Laboratory."

"Hans, how many people are in your company now?"

HG: "Six full time and two part time."

"For all three of you, is your company at this point turning a profit? If now, how soon do you expect to turn a profit?"

CT: "Yes, Kalkulo is turning a profit and has been since its first year."

HG: "That's a difficult question. We're not positive yet, but the goal is to earn back the initial investments by 2012 or 2013. As far as cash in versus cash out, we hope to be positive by 2010. It might take a little bit longer now with the financial crisis and the economic downturn.

"By the way, there are so many people now using the financial crisis as an excuse for not doing business. In my opinion, some people are getting lazy, using the financial crisis as an excuse, and that we don't like."

"Is the financial crisis also an opportunity?"

HG: "Definitely. All the investors we talk to say that downturns are the time where you should start a company. We don't have that many costs, so we don't have to cut costs at this point. A lot of other companies do have to cut costs. We can use the financial downturn to find our position. If we manage that, we can float really well when the economy goes up again... Actually we are more positive now than we were a year ago. Of course, this is also because we have a stable financial situation."

"Viktor, is your business profitable yet?"

VE: "Yes, I think it is profitable. The business was started in 2008 and we haven't finished the fiscal year, but we have income and things are looking good."

"Are there ways the Simula culture helped you get started, or it could not have been done without Simula?"

HG: "One reason I actually came to Simula was that I heard there were opportunities for starting businesses. If I had to quit my job to start a business, maybe I would have never done it. When someone tells you that you can do a startup if you want to, that is a very good thing."

VE: "As Hans said, they encouraged people to look for commercial potential. That is a starting point. However, you should consider the price of trying to do both commercialization and research at the same time. Does it affect the research in any way? As an example, it's not obvious how to collaborate with others in other institutions, if you are always looking for ways to exploit things commercially. This is something quite new in Norway, which started around 2003. All the universities in Norway got technology transfer offices as well. It's been quite a short period to evaluate how it works and the consequences, not only on the commercial side but also on the research side. So it requires some serious thinking. I don't have the answers."

"Finally, where do you see your businesses in five years?"

VE: "There is a lot of competition internationally, so to be successful we need to grow. Currently we are three full-time people. In order to get more customers we need more people."

"Do you have a target?"

VE: "We will have tens of people, without being too specific."

"Hans, where do you see yourselves in five years?"

HG: "In five years, we have not been sold to anybody, and we have grown our own company ourselves. Our platform is the preferred solution for building, sharing and running all kinds of project processes. We are the infrastructure, a more or less pure technology provider that provides a platform for different types of uses. There are companies doing software development and process development on our platform. We hope that in five years there is an ecosystem around Symphonical.

"That is how I think the Web will develop: more platforms, more ecosystems, more tightly integrated. We have to find a position, but we think platforms are the right way."

VE: "Can I add a few words? We decided early on that we would not like to be a consulting company, because that's not our goal at all. Rather, we are developing our own software that we will sell. As I have mentioned, we have this core technology for streaming, and on top of that you customize it for different application areas: video surveillance, conferencing, live broadcasts, or whatever."

CT: "Strategic decisions are very important, and I think we are in a position where we should take them seriously now. In five years I hope Kalkulo is more widely recognized, and we have visibility as a company for quality and delivery within the competence that we possess now. We should also grow. We now have the StatoilHydro collaboration; we should seek one or two similar collaborations or projects to get two or three really good customer relationships. As part of this strategic discussion, we will continue as a consultant, but it is also possible for Kalkulo to look for more product opportunities."

Epilogue

It seemed only fair to give Audun Hansen, the current CEO of Simula Innovation, a chance to comment on some of the issues raised in this discussion. Hansen agreed that in some ways Simula Innovation is still learning. For instance, Viktor Eide's feedback has helped them see the need for a more explicit list of services. One particular concern, Hansen says, is that Simula does not have a mechanism to help a new company get past the "intermediate phase"—after they have received a validation grant, and confirmed commercial potential, but before the product or service is quite ready to come to market. This is the stage where an "incubator" might be useful. Unfortunately, Hansen says, Simula Innovation does not have sufficient resources to serve as an incubator, only as a sort of "pre-incubator." For that reason, he hopes to find an outside partner, such as IT Fornebu, that would be willing to take on this role.

42

SPINNING OFF FROM SIMULA

Are Magnus Bruaset, Viktor S. Wold Eide, Frank Eliassen, Hans Gallis, and Christian Tarrou

Abstract Innovation is one of the three pillars of Simula Research Laboratory. In particular, the subsidiary Simula Innovation is dedicated to identifying research results with commercial potential and assisting the researchers in their initial steps towards business creation. However, history shows that the routes taken from the initial idea to an up-and-running company can be very different from one prospect to another. By sketching the stories of three companies that have spun off from Simula, we illustrate the diversity in the business creation processes and the need for an adaptive approach to bringing research to the marketplace.

Are Magnus Bruaset
Simula Research Laboratory

Are Magnus Bruaset · Frank Eliassen
Department of Informatics, University of Oslo, Norway

Viktor S. Wold Eide · Frank Eliassen
Lividi AS, Norway

Hans Gallis
Symphonical AS, Norway

Christian Tarrou
Kalkulo AS, Norway

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_42,

Innovation and Business Creation at Simula

Just as in other top international research institutions, life at Simula is characterised by frequent encounters between creative minds. Such encounters stimulate curiosity, give birth to new ideas, and generate energy. Although many, if not most, of these impacts fade out like dying supernovas, some of them lead to new projects and boosted activity. The result usually takes the form of a traditionally designed research project, but once in a while scientific curiosity is paired with entrepreneurial instincts. This combination can prepare the ground for new business opportunities.

Since Simula Research Laboratory creation in 2001, seven spin-off companies have been established. These companies have resulted from different processes and under different conditions. One of these seven companies, Kalkulo AS, is a subsidiary of Simula, while the others also involve external owners.

Simula Innovation AS, Simula's instrument for innovation and business creation, assists in the build-up of new companies based on the research performed in the laboratory. Simula implements an organic policy, in the sense that it is natural to set the spin-off company free if or when its business grows in a direction outside the laboratory's fields of interest and competence. On the other hand, Simula takes an active part in the development of those companies for which its expertise is seen as useful.

The following sections present three distinct business cases that have led to the creation of spin-off companies: Symphonical AS, Kalkulo AS, and Lividi AS. In addition to illustrating the wide range of possible paths that can lead to business creation, these examples also span all three research departments at Simula: Software Engineering, Scientific Computing, and Networks and Distributed Systems. For a supplementary look backstage of these three companies, please see the interview with each group's respective directors on page 601. In addition, more information about Simula's industrially oriented activities can be found in chapter 38 on page 533.

Business by Coincidence: Symphonical AS

In 2004, Hans Gallis was a PhD student in Simula's SE department, investigating methods for process improvement within the information technology (IT) industry. For one of his empirical studies, he was an observer in the technical meetings of one of the SC department's software-oriented project groups. In the aftermath of these meetings, he coincidentally entered into discussions with two members of that project group, Åsmund Ødegård and Ola Skavhaug. These discussions concerned an already existing approach to how teams could organise ideas and other bits of a project's information cloud by placing Post-it[1] notes on the walls. How could this highly visual and flexible information management strategy be transferred to a digital environment? Hooked on this idea, the two developers spent several hours of their spare time implementing a web-based prototype. After this first prototype, Simula funded a summer internship to develop a more mature version of the application.

[1] Post-it is a registered trademark of 3M.

The team immediately hired Magne Westlie, who had then just finished his master thesis and would later become Symphonical's first employee.

The wheels turned faster and faster and in the autumn of 2004, Symphonical became a prioritised project at both Simula Innovation and Birkeland Innovation[2]. The additional funding allowed the project to buy 40 per cent of Gallis' time and to hire two more developers, Terje Torma and Jørn Gabrielsen, throughout most of 2005. During this period, the third prototype was developed.

In the autumn of 2005, a few business angels were recruited from Gallis' personal network of contacts. One of them was Bjørn Haugland, a serial entrepreneur who, among many other ventures, founded the IT company Confirmit. With the support of Simula Innovation, Haugland, and other investors, Symphonical AS was established as an independent company in December 2005. By then, the company had already moved from Simula's offices to an old building located in downtown Oslo.

So, what happened with Gallis' research? The urge to obtain a PhD degree faded as the opportunity for creating a business unfolded and 2009 will be Hans Gallis' fifth year as CEO of Symphonical AS. Based on a knowledge of work processes but kick-started by an unforeseen meeting between three researchers, the development of this spin-off company demonstrates that business can be created 'by coincidence'. In the next paragraphs, Gallis gives his view on the development of Symphonical and comments on some of the lessons learned.

Characteristics of the Optimal Software for Process Management

Most of the project management tools today are almost identical, even if there are hundreds of them. When starting to develop Symphonical, the first question to answer was why are there so many identical project management tools on the market today? The main reason is that the process or the methodology (for instance a software development process such as Rational Unified Process or SCRUM[3]) is built into the tool. Following such a principle, any change to the process also leads to the development of a new tool, which is a huge waste of time and resources. These project management tools tend to be inflexible and are hardly used by professional project managers.

Examining a selection of standard tools for project management, one observes that they are based on the same traditional elements. For example, users are usually limited to creating activities, tasks, or milestones. So how should they handle ideas, user requirements, user stories, SWOT analyses, evaluations, brainstorming sessions, and all the other pieces of the information puzzle? Well, these important aspects of the work process are usually documented in a spreadsheet, a text document, or some presentation slides or scribbled on a stack of Post-it notes. All these information carriers are outside the project management tool. Consequently, the next question was what are the most widely used tools for project management? Our own

[2] Birkeland Innovation is the technology transfer office at the University of Oslo.

[3] Unlike many other terms in an IT professional's vocabulary, SCRUM is not an acronym but refers to an abbreviated form of the verb *to scrummage*, which is to restart a game of rugby after the ball has been out of play.

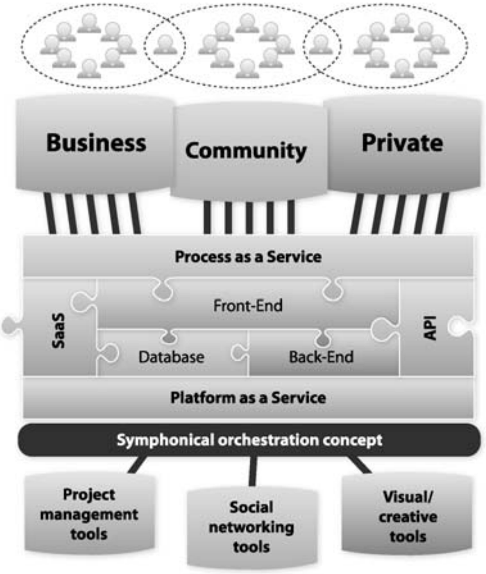


Figure 42.1 The structure of Symphonical's technology.

research points to email combined with spreadsheets, presentation slides, and text documents. Spreadsheets and email. however, were not developed to handle collaboration in a very effective way and they were never intended to do so. We have all, professionally or privately, experienced how messy email can be when trying to coordinate tasks and activities. Even worse: Who is in charge of the latest version? Who has received the information? Which tasks have been done by whom? Email is collaboration chaos! However, everybody has an email account and knows how to use it. Consequently, all project members know how to use the technology. The same observation applies to spreadsheets. Most people know how to use it to produce all kinds of tables, although a lot fewer know how to handle the real power of spreadsheets. That is the strength of spreadsheets: Their flexibility puts the users in control of almost anything they want.

In summary, the mission for Symphonical was to define a platform in which the users could create any type of process, both from scratch and from a predefined process template, while keeping a low user threshold for email and the flexibility of spreadsheets.

A Web-Based Platform for Process Management

Symphonical is a web-based platform for building and running any type of process, which competes with hundreds of existing, traditional project management tools. However, these tools can be viewed as one single competitor, since they are almost identical, technically and business-wise. In addition to the existing tools for project management, the plain vanilla combination of email and spreadsheets offers some of the strongest competition.

Figure 42.1 depicts the structure of Symphonical's technology. To create a truly disruptive innovation, one that can compete with existing project management tools as well as email and spreadsheets, Symphonical combined the best of the breed of project management tools (flexibility), social networking tools (collaboration), and creative/visual tools (usability). This foundation created the concept of *orchestration,* in which the user can build and manage any type of process from scratch or from predefined process templates.

Independence between the process and the methodology is achieved through the concept of notes on the walls of a project room. In Symphonical, you can create a process by starting with just one single note. You post the note on a wall. Then you add more notes. After a few steps, you have a structured set of notes on the wall. Then you gather all your notes and categorise them into named groups, or columns, such as "great ideas", "good ideas", and "poor ideas". Consequently, on this wall, a note is an "idea". Taking a step back for an overview, you decide that you need another wall where you can set up activities needed to implement the ideas. On this wall, the notes are turned into "activities". In this way, the user can define the purpose of each note. In addition, the user can put more process elements into the tool by structuring the wall (for instance by categorising the notes into columns) and then by putting together a set of walls. The result is a user-created virtual room with several walls, where each wall has its own individual structure and notes. The user has created a process from scratch, implicitly through the way the user and his or her colleagues work. An implicit process has become an explicit one that can easily be shared with other users or reused as a process template. We like to think that Symphonical has invented process as a service!

Symphonical's Approach to the Market

Symphonical offers a web-based process-independent collaboration platform in which the users can create and manage any type of process without any programming knowledge. The users start from scratch, typically by turning implicit processes into explicit ones. This transformation makes it very easy to reuse and share process knowledge.

In this way companies can save considerable expenses since they avoid the need to buy a new software tool every time they change their processes; they just create a new process or adapt their existing processes by using the same tool. Moreover, the cost of running processes decreases dramatically since the new technology can handle unlimited types of processes. This cost is only a small fraction of the costs of today's existing PM tools.

Symphonical aims at earning money from partners who create content (processes) and sell Symphonical's services to their customers, as well as from end users (organisations or individuals) who buy services directly from the company or pay by viewing advertisements.

Today, Symphonical employs eight persons, of which six work full-time and two are part-time. The company recently signed agreements with its first partners. Consequently, the first income is based on direct sales. Symphonical's strategy is to build an ecosystem consisting of both content and technology partner networks, which will grow our business model. Hopefully, this partner strategy will blossom during 2009.

Experiences to Be Shared

Symphonical has greatly benefited from meeting a true business angel, Bjørn Haugland, who has invested time, resources, and money in the company. The development of the first official version of Symphonical started in January 2006 and the alpha version of the platform was released that autumn. Since then, the system has gone through a repeated cycle of development and testing. During 2007, the team met several obstacles but released a private beta version during the first half of year. The public beta version was released in July 2008. Looking back, the development of the web-based platform concept turned out to be very demanding in terms of both time and resources, thereby causing the delay of releases.

From spring 2004 until the end of 2008, Symphonical spent approximately 10 million Norwegian kroner. This funding stems from Simula Innovation, Birkeland Innovation, Innovation Norway, the Research Council of Norway, the Nordic Innovation Centre, and private investors. Approximately 50 per cent of the total funding stems from business angels.

Three major experiences should be of interest to other research-based entrepreneurs:

Stay in there, just like you do with your research! Keep focused and stay focused for a very long time. Developing a disruptive technology takes years of hard work. Do not spend all your energy during the first months. You have much more time than you might think. And if somebody reaches the goal before you, there is nothing you can do except create a new strategy.

Involve your network, colleagues, friends, and family! You cannot do this alone. Create a company with a board of directors that includes external people with lots of experience (especially entrepreneurial), a large network, and some money. Choose the people you like. You are creating your own company and you can now decide with whom you want to play. I suggest playing with smart and fun people, because it will be a very long journey.

Money should think not talk! In an early stage, involve business angels with enough money for you to get started. Often there will be need for a couple of years of funding without any income for the company. Business angels are great assets, particularly in their ownership of the money. Since they put their own money on the line, they do not have to defend how it is spent. This means less paperwork and bureaucracy and more time to focus on your idea, product, and company. Also, you can count on gaining access to their experiences, both positive and negative, as well as their knowledge.

Business by Necessity: Kalkulo AS

In 2005, Simula entered a research collaboration with the oil and gas company Hydro, in which the laboratory's special competence in applied mathematics and information technology was paired with the oil company's expert knowledge of geology and geophysics. After the merger between Hydro and Statoil in 2007, this collaboration was brought forward and further strengthened in the new company StatoilHydro. Further details of the collaboration scheme and the scientific contents of the work are presented in chapter 40 on page 553.

From the very beginning it was clear that the research collaboration would also require the delivery of software tools for use in the oil company. This need led to a general discussion of whether Simula should collaborate with industry in projects that require not only research but also deliverables of ready-to-use technology. At Simula Research Laboratory's general assembly in 2005, the Norwegian government explicitly expressed, in its capacity as the majority owner of the laboratory, that such collaboration was not only allowed but desired. The turmoil behind this decision is described in chapter 7 on page 69. In addition to supporting Simula's intention to enter into a long-term industrial collaboration, the general assembly concluded that the laboratory had to implement procedures that would keep the funding of basic research activities separate from the money flow governing product-oriented development.

In parallel with the discussions concerning Simula's industrial involvement, activities within the collaboration were planned. It soon became clear that important activities would call for high competence in advanced visualisation and geometrical modelling. At the time, this was available in the SC department through two scientific programmers, Christian Tarrou and Trond Vidar Stensby, and one guest researcher, Øyvind Hjelle. However, a practical problem arose as these persons were hired on temporary contracts that soon were to expire, and permanent positions in Simula could not be offered.

Motivated by the need to separate the funding of technology development tasks from basic research and keeping expert competence in-house, a simple and effective solution was devised. In April 2006, Kalkulo AS was established as a commercial subsidiary of Simula: Business was created "by necessity". This company has its own economy and is allowed to take on commercial contract work, including consulting. The company is still, however, co-located with Simula and benefits from full integration with its parent organisation. Christian Tarrou, a member of the initial team, took over as director of Kalkulo in 2007. In the following paragraphs he gives his view of the company and its future development.

Getting Off the Ground

When Kalkulo was established, the initial team consisted of four scientific programmers and the company's director. Before the start-up, the team's expertise in scientific visualisation and 3D geometric modelling was used in internal projects in Simula, for instance, supporting the cardiac computing project in the SC department. In addition, the team also represented substantial experience with large-scale software

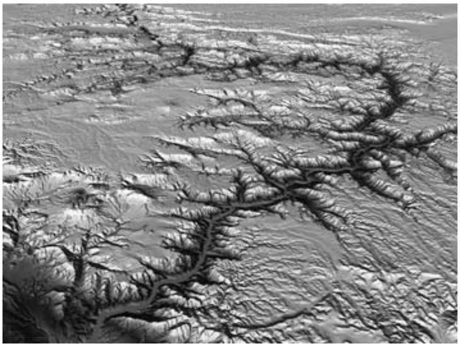


Figure 42.2 This digital elevation model of a canyon is based on high-resolution terrain data. The realisation of such a scene requires Kalkulo's compentence in geometric modelling and visualization. Advanced multiresolution algorithms are employed to ensure real-time rendering of the scene, and sophisticated visualisation techniques help enhance the very finest details of the terrain.

development, especially in the context of scientific applications. This competence, which is still a trademark of the company, has been built over several years of work in research institutes and the software industry, see figure 42.2.

From the beginning, it was clear that the collaboration with Hydro would call for Kalkulo's special competence. In particular, Hydro wanted to realise a software application capable of visualising geological data independent of spatial scale and across geological time. Such an application, supposed to visualise huge data sets ranging from a very global scale down to the finest details, called for advanced multiresolution display techniques. This application was the idea of Jakob Skogseid, Hydro's first project leader for the collaboration. He had actually had this idea for several years, without a way to realise it, when suddenly the access to Kalkulo's competence and professional experience made it possible. Today, the resulting application, commonly referred to as the 4D Lithosphere Model, is used by StatoilHydro and select partners.

Kalkulo has also been instrumental in another type of technology development requested by Hydro. The challenge offered to Simula and Kalkulo was to enable the Compound Modelling technology to work in three spatial dimensions. This technology, developed by Steen A. Petersen at Hydro's research centre in Bergen, was already available for modelling vertical geological sections. However, the extension from 2D to 3D represents an enormous increase in computational complexity. At the time, Hydro did not have the competence needed to deal with this challenging task. This shortcoming was remedied by Kalkulo's implementation of very efficient algorithms that allow the necessary core calculations to be performed in 3D. The new generation of compound modelling is now fully operational and also includes the capability of producing geological models of much greater complexity.

Even though the collaboration with Hydro represented a flying start for the company, Kalkulo has been vulnerable due to its small size. The primary objective of the company is to secure commercial success. Therefore, the first short-term goals were focused on economic growth by operating with healthy margins. It was decided early on to operate as a consulting company, which in the context of a knowledge-driven industry is believed to limit the risk, compared to being a product vendor. Furthermore, Kalkulo decided to keep its focus on the existing core competence and to seek new projects using existing networks and very targeted marketing.

Kalkulo Today

As of January 2009, Kalkulo has been in business a little less than three years, during which we successfully operated as a consulting company. The economic results for these three years all show a healthy profit. At the same time, the staff has expanded and currently counts nine full-time employees. In other words, the initial short-term goals of securing economic growth have been reached. This growth was realised within the company's initial domain of competence.

Our business idea is to deliver tailored software solutions for technical applications. There are two key components that form this competence: a thorough scientific and mathematical background and extensive experience in the development of industry-standard software. So far, we believe that an important reason for Kalkulo's success is that we are able to combine these two skills. We also feel confident that the future for this kind of competence is good. The amounts of data coming from various sources are increasing, and new devices for acquiring data are emerging. All in all, there is an increasing need for software that can present data in a way that helps the user sort the information from the noise. The ability to co-visualise different data types is an important part of this task. We therefore believe that Kalkulo's special competence in scientific visualisation will find a wide range of applications in the years to come.

Kalkulo is in the process of developing an internal technology platform, which is built on top of software components already developed in the company, as well as on select third-party tools. This platform is designed to support the implementation of software applications for the visualisation and presentation of a wide range of data types. Moreover, this technology is independent of specific application areas and can be reused in very different settings. We believe that this platform can provide a basis for a multitude of projects, ranging from the rapid prototyping of research ideas to the construction of complete end-user applications. Representing a generic framework, the technology platform has the potential of being a starting point for developing stand-alone products once good product ideas are identified.

We have already emphasised the importance of the projects in the StatoilHydro portfolio. This collaboration is organised so that the projects are headed by the Computational Geosciences group at Simula. Kalkulo is involved in this activity as a subcontractor and participates in the function of a consulting company. We take care of the development part of the collaboration, thus supporting the research part performed at Simula.

The project that Kalkulo runs for the Inverse Problems (IP) group in Simula's SC department has been an important contribution to the company's strategy. This activity was partially funded by the Research Council of Norway. The IP group received funding through a special programme providing support to projects that aim at verifying the commercial potential of research results. One important research result from the IP group is an advanced algorithm for the estimation of myocardial ischemia, see chapter 22 on page 287. Ischemia is a reversible condition that precedes cardiac infarction. The goal of the verification project was to test the algorithm on data from real patients. These data sets included medical imagery of the torsos of different patients. One of Kalkulo's roles in this project was to construct patient-specific 3D geometrical models from such images. Another contribution from Kalkulo was the construction of a software tool that offers a "virtual laboratory" for studying the sensitivity of the algorithm with respect to perturbations of the patient's geometry.

In conclusion, the existence of a consulting company within Simula has been a benefit for both projects mentioned above.

Kalkulo's Future Role

The current situation, as described previously, represents only the first steps towards the role that Kalkulo will play in the future. One of the goals for that role is the ability to serve Simula within all its fields of research. At present, Kalkulo is collaborating only with parts of the SC department through the Computational Geosciences and Inverse Problems groups. Consequently, in order to reach this goal, we must find projects that involve the ND and SE departments as well. We believe that the results obtained in the existing projects show that this goal is worthwhile.

In order to serve projects together with ND and SE, we need to build up new and relevant competence. A natural way of doing this would be to recruit personnel from Simula when specific projects are initiated. We believe that developing a wider competence, and thereby representing a broader selection of Simula's research activities, may have even more benefits. In that case, it would be natural for Kalkulo to look for industrial projects that require our total competence, which of course would increase the number of possible projects we are able to take on. We would also be able to take on larger and more complex projects. A broader competence within Kalkulo would also represent a strengthening of our development team. In turn, if Kalkulo is able to contract new projects that require competence from more than one of the research departments, this might trigger collaboration between the corresponding research groups.

All in all, we believe that Kalkulo should remain co-located with Simula, since physical presence is a considerable strength in the continued development of the relationships between the companies. This relationship should develop into a collaboration on different levels, namely, research and development projects, competence, personnel, and networks of contacts.

Business by Research: Lividi AS

As the Internet and other communication networks have become integral parts of people's lives, the demand for higher quality and robustness of the services delivered over these networks has increased. Between Simula and the University of Oslo, basic research on the adaptive streaming of real-time media has been conducted for several years. In this context, the term *adaptive* refers to the streaming software being continuously aware of the environment within which it works, such as the available network bandwidth and the performance properties of the end user's device. In particular, the technical solution emerging from the research is capable of offering these levels of adaptivity in a cost-effective way.

The company Lividi AS, which was established in 2008, has grown out of the basic research activities, partly based on the PhD work of the company's CEO, Viktor S. Wold Eide. In many respects, one could claim the route taken in creating Lividi is a textbook example of how to bring research to business: from PhD theses and scientific publications, via innovation projects and prototyping, to a solution offered to the international market on commercial terms. From this observation, this spin-off has shown how business can be created "by research".

Frank Eliassen is a professor and leader of the ND research group at the University of Oslo and also a senior researcher and project manager in Simula's ND department. He has conducted research on distributed systems since the early 1980s and research on multimedia streaming since 1994. He is one of the founders of Lividi. In the following paragraphs, Eide and Eliassen explain the underlying business idea and comments on the path taken from research to the company's creation.

Lividi's Business Idea

Lividi provides an Internet distribution platform that offers cost-effective, optimised distribution of live video content. As a result, end users can experience an improved and personalised viewing experience with fewer interruptions and less distortion. In addition, content and service providers will have more satisfied customers, who in turn will want to access more video content. Thus, the business proposition is enhanced for both customers and providers.

The Lividi solution can be integrated and customised into end-user devices or network elements such as base stations or server-based infrastructures. Our software automatically adapts a video stream to individual device characteristics, varying network quality, and user preferences. At the heart of this solution is an innovation based on a combination of results from research undertaken at Simula and the University of Oslo since 1998. These research components include *scalable video coding* that provides flexibility for streaming over a wide range of bandwidth availabilities, *automatic control of adaptation* that usefully exploits this flexibility while taking user preferences into account, and *an overlay distribution infrastructure* tailored to transport scalable coded video data and that efficiently disseminates live video to large numbers of Internet users.

A Solution for a Market in Pain

Watching video on the Internet has not yet lived up to expectations. Users suffer from low-quality video, long delays, distorted or frozen images, choppy frames, and other problems. In sum, the video experience is not satisfying. Obstacles to improving the quality of current video streaming solutions include the heterogeneity in end-user devices, ranging from mobile phones to home cinema theatres, and the different underlying networking technologies, including 3G, Wi-Fi, ADSL, and cable access networks. Also, the bandwidth available for transmitting video varies dynamically due to the best-effort nature of the Internet and changing channel conditions, especially in wireless networks. Today's video streaming solutions cannot properly cope with such diversity and variations.

Lividi offers video-streaming solutions for content providers and network operators. This includes customisation and integration of software into their video distribution networks, as well as support and enhancements of the functionality over time. The company bases its business on a video-streaming system layered on top of an efficient many-to-many distribution technology, which allows for fine-grained and independent personalisation and adaptation of the delivered media streams. A novel adaptation scheme takes advantage of the fine-grained personalisation and adaptation opportunities provided by the underlying scalable video-streaming system. The Lividi software platform adapts and maintains multimedia applications and services in response to dynamic changes in network quality and user preferences in such a way that the user's quality requirements are satisfied. Instead of fixed-quality video, Lividi streams are adjusted to the client's preferences. Our technology provides the flexibility and adaptivity required to handle Internet video streaming in a scalable and cost-effective way. This flexibility, combined with excellent scalability and low server costs, makes it suitable for any live media application, on any platform.

A Business Model Providing Customised Offers

Lividi provides adaptive video streaming solutions geared towards a rapidly growing live video-streaming market. This adaptive video transport improves customer value through improved resilience and graceful degradation of video quality. Our offer includes customisation and integration of software into the customer's video-streaming applications, as well as support and enhancements of the functionality over time. The offer combines an initial specialisation and customisation project and continued licensing. For video-streaming system integrators, the following needs are addressed:

- Video streams can be prioritised.
- Important regions of individual video streams can be prioritised.
- Priorities can change over time based on interesting events, as determined by video analytics.
- Better quality for the interesting events improves video analytics.
- Heterogeneous terminals and networks are supported.
- Variations in bandwidth and load are handled gracefully.
- There is continuity of service quality even when bandwidth availability fluctuates.
- Video streams can fully utilise spare bandwidth.

Lividi's intelligent adaptive video transport provides cost reduction and flexibility. In particular, it

- Avoids overprovisioning network and processing resources,
- Allows off-the-shelf IP-based wired and wireless network technology to be used,

- Supports efficient many-to-many video streaming, and
- Reduces development costs for system integrators.

Market Potential and Goals

The market potential for large-scale video streaming is substantial and growing quickly. Gartner estimates the number of IPTV subscribers will reach 48.8 million by 2010. In this context, Gartner defines IPTV as *"video programming (either broadcast or on-demand) over a carrier's managed broadband network to a customer's TV set."* Global IPTV revenue is expected to reach USD 13.2 billion by 2010. Lividi targets a broader market than plain IPTV, also including PCs and laptops, mobile TV, and Windows Media Center, Xbox, and PS3 platforms[4], among others.

Even as companies are actively entering this market, offering solutions for different types of customers and media, major hurdles related to the quality of the viewing experience and the cost of distribution still hinder mass-market diffusion.

In late 2007, the Scalable Video Coding standard (H.264/SVC) was finalised by the international standardisation bodies ITU-T and ISO, with backing from the major industrial players and the scientific community. Scalable Video Coding allows for streaming video over a wide range of bandwidth availabilities, with negligible overhead.

In spite of the modest quality of experience offered by today's video-streaming solutions, video streaming over the Internet has achieved widespread adoption. Improvements in user satisfaction through Lividi's optimal and personalised viewing experience will increase usage and create new market opportunities.

Lividi has narrowed the market scope to perform the initial launch of its technology. Three segments have been identified as the most interesting and discussions and presentations have commenced for possible partners. Our technology is generic and to a large extent reusable across market segments in need of efficient and adaptable live video distribution. By early autumn 2008, Lividi signed a contract with its first customer, a significant international actor in its market segment.

In the rapidly expanding market of Internet video streaming, Lividi is targeting a growth corresponding to ten employees by 2010, with expected sales of 20 million NOK.

Lividi's Current Status and the Road Ahead

Lividi is a spin-off from the ND groups in the Department of Informatics at the University of Oslo and at Simula. The company is owned by the founders together with Simula Innovation and Birkeland Innovation.

The Lividi venture was founded in the spring of 2007, when the first round of funding, two million NOK, was secured from the Research Council of Norway. Additional initial funding was granted from Innovation Norway, Simula Innovation, and Lividi's participation in the two international research and innovation projects MARCH and

[4] Windows Media Center and Xbox are registered trademarks of Microsoft Corporation, while PS3 is a registered trademark of Sony.

VERDIONE. This participation is partly funded by the Research Council of Norway. Lividi also recently received an internationalisation scholarship of 500,000 NOK. The first phase of development was aimed at verifying the technology and was successfully completed by building a demonstrator. This phase was also used for customer verification. which allows us to improve our understanding of the needs of potential customers and to establish initial contacts with potential seed and venture capital investors.

Our next major step of development is aimed at a successful implementation of the Lividi technology in one or two pilot customer projects. These will involve the customisation and development of solutions that are not in conflict with our longer-term ambitions, as well as extraction of reusable solutions and knowledge from customised products. To this end, Lividi signed a contract with its first customer. We are currently also establishing and maintaining contact with potential seed and venture capital investors, as well as developing further business plans and products for entering additional market segments, as indicated previously. The goal is to sign further customer contracts. This work includes more attention towards international markets and collaboration partners.

The Lividi team consists of eight persons, five of whom work for Lividi on a daily basis and three who are full-time employees.

The Role of Technology Transfer Offices

The support and enthusiasm of Simula Innovation has been very valuable in Lividi's establishment phase. In particular, the T2M process and the provision of business development expertise assisted in establishing the initial focus of our endeavour: developing a business plan and talking to potential customers and investors to better understand their concerns and needs. In total, these steps have enabled us to formulate a more precise business proposition.

It would seem natural in an early phase of the company's development to expect a significant contribution to the planning and the operation of the business from large (institutional) owners such as Simula Innovation and Birkeland Innovation. To live up to such expectations, it is important that appropriate resources be allocated to the tasks at hand. We believe this point should be taken into account when considering the future goals and strategies of technology transfer offices such as Simula Innovation and Birkeland Innovation.

43

WE'RE NOT A TELCO, WE'RE A WEBCO

An interview with Olav Lysne and Hans Christian Haugli by Dana Mackenzie

One of Simula's newer industrial partners, Telenor, is the seventh-leading provider of wireless services in the world, with an existing customer base of 170 million users and a presence in 30 different countries around the world. Since 2007, Simula and Telenor have jointly funded a project called Simtel. The project, organized within Simula Innovation, now includes five full time employees and with a sixth employee starting soon.

Although the Simtel project lies within applied research, parts of it clearly have their intellectual roots in work that was done in Simula's Networks and Distributed Systems department, dating back to 2001. At that time, Olav Lysne, now Director of Basic Research at Simula, began a project called ICON dedicated to "interconnection networks," such as the supercomputers or clusters of computers. The research was intended to eliminate problems such as deadlocks, which occur when each processor is waiting for a response from another one.

However, the research had an unplanned benefit. Some of the same ideas could also be used to re-route data packets when part of the network broke down. As a result, the ICON architecture has been adopted enthusiastically by industry, including such clients as Sun Microsystems, Silicon Graphics, and Sandia National Laboratories. Ironically, says Lysne, what has attracted these customers to ICON is its fault tolerance–which is not really what it was designed for.

As important as fault tolerance is for supercomputing clusters, it is even more critical for the Internet and for wireless networks, where parts of the network go offline on a regular basis. Thus, the ICON research positioned Simula perfectly to study fault-tolerant architectures in these networks. A second project, called Resilient

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6_43,

Networks, was launched in 2006, also within the Networks and Distributed Systems department.

As it turned out, Simula was not the only organization interested in resilient networks. From Telenor's point of view, we are now entering an age where the separate networks of the past–wired phone networks, mobile phone networks, wireless computer networks, and the Internet–will all be linked into a single cohesive unit via IP, the Internet Protocol. Hans Christian Haugli, Director of Research and Innovation at Telenor, puts it this way: "I don't like to call my company a telco, I like to consider ourselves a Webco." In this new IP world, resilient networks may be a crucial ingredient to improve the customer's experience.

We asked Lysne and Haugli to talk about the Telenor-Simula collaboration, and what benefits they see in this merger of Telenor's vision with Simula's technical expertise.

"Hans Christian, how did the collaboration with Simula start, from your point of view?"

Hans Christian Haugli (HCH): "I was on an industry committee on which Anita Krohn Traaseth[1] was also serving. We had some discussion of innovation in that committee. She expressed attitudes that were similar to the type of attitude we had at Telenor, and for that reason we ended up talking during breaks. I had been outside of Norway for 25 years, so I didn't know very much about you guys."

Olav Lysne (OL): "At the time, we were quite new. The connotation of Simula was the old Simula programming language and not the research lab. So it's not surprising that when you came back, Simula was *terra incognita* for you. I think it's true that Anita really opened the door here."

HCH: "Then, about two summers ago we decided to have a chat about further cooperation. We brought the two management teams together on a shrimp cruise, and talked about research and innovation. Kristin Vinje was there, and Aslak Tveito. I thought, 'These guys have the right attitudes! I can do business with these guys!' I've talked with a lot of people from other academic institutes who have very different attitudes. Simula demonstrated an attitude, not just to be probably the highest ranking academic institution in Norway in its field by a large margin, but to recognize that the world was about more than publication. They were willing to take steps to see their results being used in practice."

OL: "That is really important. When we started Simula, we looked at other institutions and tried to see what we could learn from them and what we could do differently. The attitude towards making a difference outside academia was one thing that we really wanted to do differently."

[1] Director of Simula Innovation from 2005 to 2007.

"Olav, you have written about the willingness of Simula to look at the long time frame. Does that also distinguish Simula from other institutions?"

OL: "Many institutions look at the long time frame. What I would consider a distinguishing feature of Simula is that we sustain the time frame. We see that the application may be many years ahead, but as long as the project remains in a position where it is still promising and is moving toward this application, we sustain it and sustain it. But we do not allow projects to stay ten years away from an application. If five years have passed and you're still ten years away, then something should be done about it.

"My ideal of a Simula project would be that it starts as a forefront project with a significant and very clear taste of basic research, but it has a life evolution from oriented basic research to applied research and then we go the extra mile to see that it has an impact outside academia."

HCH: "The willingness to go that extra mile, to bring your competence out into the world–that's important in terms of being successful and changing the world, which is what we all want to do. That, to my mind, makes you unique among Norwegian academic environments."

OL: "At least in Europe there's a gap between industry and academia that really needs to be filled. It's really easy to stay in academia and be successful, pushing out important papers. Other academics read these papers and write another paper. You can draw a circle around it, and nothing really passes that circle and gets out. That is something I think academic institutions should address."

HCH: "I think the things we do should be part of trying to make a better world. It doesn't mean we don't like business, but you need to start with something which is sustainable, something which improves somebody's life and makes something better. If you do that, then you create value, and if you create value you have a good chance of tapping into some business model and getting paid for it."

"How does the Simtel project contribute to your business model?"

HCH: "In most strategies you start out with a vision. The vision we have is that we're moving into a new world, a pure IP world. All devices are connected over IP and you have lots of applications sitting on top of them. On the bottom of this stack, if you like, you have different type of access networks, such as GSM and WiFi.

"We call our strategy the Always Best Connected strategy. In the future, the cost of making a cell phone able to use different access networks will be very small. To do WiFi on a cell phone is perhaps 3 dollars a day. What is the world going to look like when you can communicate over many networks at the same time? Then we can perhaps move from the old-fashioned paradigm, where we guarantee certain availability, which tends to be fairly costly, and move over to a more statistical approach. We'll guarantee up to a certain level. Beyond that, for *x* per cent of the time you'll pick the best network available and bring the quality and the user experience up. This Always Best Connected strategy, implemented in a transparent way to the user, is something we really believe in. The resilient networks, the competence Simula has and the algorithms underlying it, will support that strategy."

OL: "That impacts what kind of applications you put on top of the network, and what these applications need to be able to handle. They will need to take advantage of this higher bandwidth that's available. They may need to handle such things as cost issues, because Telenor is, of course, a business. If an application has this extra bandwidth available, that will probably come with costs. How do you handle that in such a way that the user has control of the cost? There's a range of questions that goes all the way from user perception to deep technical things, such as what should really happen on the protocol layer as you switch from GSM to WiFi."

HCH: "I don't like to call my company a telco, I like to consider ourselves a Webco. We aren't there yet, but that's the way we are heading. In order to increase the data usage of users, especially mobile users, the user experience is the absolute barrier. Apple has managed to reduce the user barrier, to make their products simple to use, and they are really taking off. People really like Apple's applications, across all the screens. We talk about four screens. You have the cell phone, the PC, the TV, and you have perhaps a game console. Information should follow you wherever you are, across those four screens.

"The user experience is so important. The user experience has to do with bandwidth. You need to get the bandwidth up, you need fast response and you need good quality wherever you are.

"In addition, we see that bandwidth usage is exploding. The challenge for all operators is that we generally tend to impose a flat rate, an all-you-can-eat rate on the data side. Now we have a situation where to get into a market, we need to give the users the best experience. At the same time, the usage goes up dramatically, and our expenditure goes up as we invest in more and more infrastructure. But we can't get more out of your users. So we have to be much more efficient in the infrastructure. The combination of user experience and efficiency is why this kind of know-how and vision is becoming more and more relevant."

"It may be too early to evaluate the partnership, but can you say what you've seen so far?"

HCH: "Number one, Simula has shown that the type of challenges we face are areas where they have competence. They can help us move forward. Number two, I think there's good progress in this area. But we all have to understand that things do take time. That's perhaps the dangerous thing. If you expect that you will do wonderful things in six months, that's usually not the case.

"Let me tell you an example that Telenor is proud of: Opera, a company that makes web browsers for cell phones. That group came out of a research unit at Telenor. Here we are, 20 years later, and we really start to see that they are do-

Olav Lysne and Hans Christian Haugli

ing very well in the market. They have ended up on the play consoles and the cell phones."

"What makes Opera good for cell phones?"

HCH: "First of all, it meets all the standards. It's very clean. The second thing, especially with this Opera Mini, is that they again address the user experience. They do compression and new techniques, they reformat pages so you can start looking at standard web pages on a normal cell phone. A wide web page becomes a long, long, thin one instead. They compress pictures, because you don't need megapixels on a 160 x 160 screen. They take advantage of that. Opera is an example of something that took a long while. They perhaps started out going not in the final direction, but they built competence along the way, they understood where the new markets were, they adapted and they became successful.

"The most important thing is that you try to address real problems, you build top competence, and you readjust as you learn more about the market and what the new problems are. It's not necessarily that the goal is 100 per cent defined, or that you know exactly where the goal is, but it's a general direction, and the competence is the most important part."

OL: "I think we feel relatively comfortable with the situation that this collaboration is in now. Hopefully when you come back in 15 years and interview us again, we'll be talking about a success story coming out of this. That's at least my wish. Can I promise it? Well no, you never can. But do I believe it? Yes, I think I do. I think we will have something to show for it, if we have the right vision.

"I'd like to say something about how Simula looks on this. We're an academic institution. We get some of our mission statement, if you will, from the Norwegian government. We work on solving problems, but we do not own those problems. The people who are out in industry own them. We can sit up in our ivory tower and try to think up what will be the glorious problems of tomorrow, but the ones who know that best are the people who are in the business. If I look at this in a very selfish way, the most important side of this for Simula is to get industry's input into our process of deciding what areas we should attack. That's one thing we cannot get anywhere else."

HCH: "For example, in the early days, you guys did a lot of resiliency in fixed networks. I think that when we started working closely together, we challenged you and said, 'You focus so much on these fixed networks, but the real problem in this industry is all these mobile networks.' True?"

OL: "That's why we now have a huge project on that. Yes, it's changed us."

"What makes resilience particularly hard to achieve in a wireless network?"

OL: "In a wired network it is relatively easy. If you lose connectivity, it's down to some physical thing happening. It's an on-off thing: Somebody's cut a cable or the electricity supply to a router has gone out. But wireless is a very uncontrolled environment. There is noise and reflection from buildings."

HCH: "The quality of a signal can change 100 times in value if you move half a meter."

OL: "Not only that, with wireless you often move half a meter. One of the things you would like to do with your wireless is actually to move!

"So two things happen. First, the failures are not an on-off thing any more. And the occasions of a failure that are relatively infrequent in a wired scenario become far, far more frequent with wireless."

HCH: "To give you some numbers, a typical fixed line will have an availability of 99.99... per cent, with six or even seven 9's. For a mobile link, you talk about 90 to 95 per cent. So it's vastly different, orders of magnitude different."

OL: "One approach, as Hans Christian mentioned, is that if I have a wireless link to my GSM, there may also be some WiFi available. My device might have to feel what the best connection is, and adapt to that."

"What other challenges are there to building this IP-based world?"

OL: "I can give technical details, but keeping your IP address as you move from being on GSM to being on WiFi is a non-trivial technical problem. From a high level on the network topology, it appears as if you suddenly disappeared from one point and reappeared at another point.

"The high level answer, though, is that one type of challenge is technical and one is organizational. At Simula we're experienced with the technical side, but I don't think our collaboration really touches the business side at this stage."

HCH: "To me there are three areas that are interesting. The first is this ever increasing bandwidth and the business models around that. How do you create a business model when usage goes up ten times but people end up paying the same amount? The second is the user experience, especially for smaller devices. How do you keep improving so that it becomes easy to get access to information? You want to reduce the number of buttons you have to push to get to what is relevant. We know that we lose 60 per cent of users for each time they have to push a key. People do not want to push buttons! The third, and I'm not sure of this, is the long-tail effect. Amazon has claimed that only half the books they sell are bestsellers. The rest are specialized books that relatively few people buy. I think that there is a possibility that the telco industry is purely in the bestseller business. We sell voice, SMS, and broadband, but what about these thousands and thousands of applications that people want? If you want the best user experience, you have to facilitate not only the bestsellers, but all these obscure services that people want to have. We want to make it very easy for people to make their own services and innovate.

"I'll give you one example which is fascinating. We have a large operation in Bangladesh, where they have fairly limited radio coverage. We introduced a service where you can have a ring tone that is a musical melody. If you don't answer the phone, the caller will hear a piece of music. You don't pay anything for that service. The locals figured out that if some guy had a nice song on his cell phone, they could call the number and listen to it. So people used it as almost a broadcasting service. We would never have thought of that! They would use the cell phone as a radio. The users are very creative.

"There is another example that we don't see in Europe or the U.S, but is common in Asia, called missed calls. People would place ten calls in a row to your phone, very rapidly, letting it ring once and then hanging up each time. On the display it would show you that so and so has called you ten times. By calling a different number of times, they used this as a messaging system!

"When I was in Kuala Lumpur half a year ago, I was in a busy shopping centre, and the taxi driver didn't know exactly when I was going to come out of the shopping centre and he also didn't know, because of the traffic, when he was going to be available. So he said, 'When you're outside, send me a missed call.' And a few minutes later, he was there, picking me up."

"Why didn't he want to talk?"

HCH: "Because it was free! This is an example of what I call user innovation. Maybe as an operator you don't particularly like these free services, but it increases the utility of the mobile device. You still have to buy the cell phone, and you still have to be a customer. This to me is an example of how people can be very, very creative."

OL: "It's extremely hard to guess how something will be used until it's deployed in society and people start to use it. That's similar to what happened with the SMS service when it first appeared. At first it was introduced as a free service. No one anticipated the volume of traffic that this service would create."

"Where do you see the Simtel collaboration going in the future?"

HCH: "Earlier I mentioned realistic expectations. We know from the venture capital industry that if you have ten projects, five or six will be dogs, total wastes. Perhaps two or three may kind of break even. One (if you're lucky) or two (if you're very lucky) will be total successes.

"What we're doing here is in an even earlier phase, so I think the probability of success is not going to be any higher. So first, we need to have a reasonable volume. Also, I'd like to say that when we do fail, we should fail cheaply. Perhaps most of our experiments are going to fail, but some will work and there will be some very big ones out of that. We are taking a statistical approach to it."

OL: "The notion you introduced of failing cheaply is important for just that reason. You can be as clever as you like in picking out a portfolio of projects. There will always people asking, 'Can't you just do better at picking the successful ones?' But history shows that this has never worked. No one has ever maintained a higher hit rate than Hans Christian just said. So the portfolio needs to be wide and failures need to be cheap. I've never heard this notion before, but it's a good one."

"Does Telenor do any basic research itself?"

HCH: "No. Generally we do applied research. We practice what we call innovation-driven research, and not research-driven innovation. The order is no accident. We like to look at an opportunity or a problem, and ask, 'How can we make a solution?' We will find there are elements missing in this solution. Then we will apply research, to look for the missing elements. As you know, 99.5 per cent of the global research is outside Norway. So we have to go outside, and find partners who have competence that we can work with. We find people who have the best competence, and I think that is what you build with your basic research. You may not know exactly how this competence is going to be used at the time you build it, but... we will find you!"

"Is there anything else you would like to say about working with each other?"

HCH: "In addition to what I call the competence at Simula, basically they are very nice to deal with!"

OL: "I can return that. I think we've had a challenge, coming from academia, and we were lucky with some of the recruitments early on, which changed us and helped us better understand how to sustain a collaboration like what we have with Telenor. I think that Simula has learned a lot."

A

SIMULA FACTS

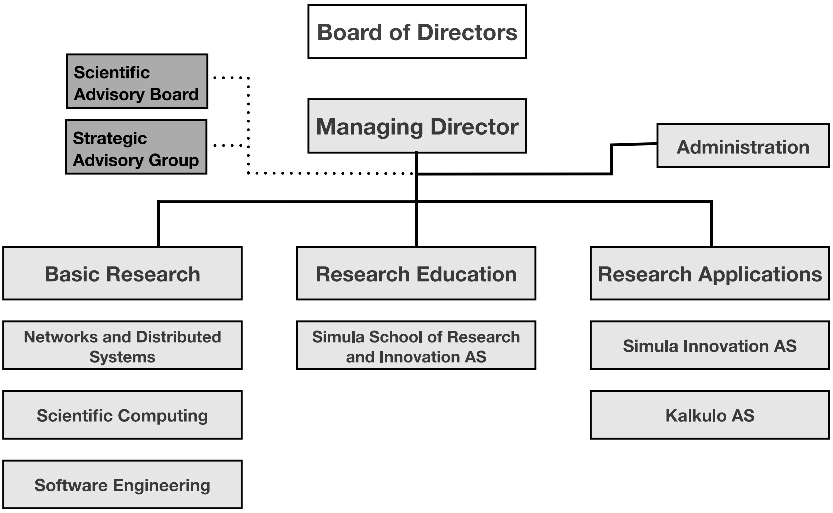


Figure A.1 Simula's organisation.

The highest body at Simula Research Laboratory is the board of directors, which is appointed by the owners of Simula at the General Assembly. Adhering to the provisions of the Companies Act, this board makes strategic decisions and approves the budget and annual reports. It appoints the managing director, who in turn organises the company's activities. The company is divided into three units corresponding

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6,

to Simula's three main tasks: Basic Research organises research activities, Research Education is administratively responsible for the PhD students and postdoctoral fellows, and Research Applications is responsible for promoting the application of the research results.

Each of these units has a unit director and, with the managing director and the director of the Administration unit, the directors constitute Simula's management group.

Basic Research is divided into three research departments, each with its own department head. All full-time researchers and project leaders are employed here.

Research Education streamlines the formation of PhD students and postdoctoral fellows. It consists of the Simula School of Research and Innovation and with Simula as the majority stakeholder. A PhD student at Simula will be affiliated with SSRI and will report administratively to this unit. The student's day-to-day work, scientific research, and supervision, however, take place in the corresponding research department in Basic Research.

Research Applications consist of the two wholly owned subsidiaries Simula Innovation and Kalkulo. There are two advisory bodies reporting directly to the managing director: the Scientific Advisory Board, and the Strategic Advisory Group.

Ownership, board and management[1]

Ownership

Simula Research Laboratory is a limited company jointly owned by the Norwegian government (80%), Norwegian Computing Center (10%), and SINTEF (10%).

The Board of Directors

The board of directors consists of the following members:

- Ingvild Myhre, Chair of the Board
- Anne-Brit Kolstø, University of Oslo
- Gunnar Hartvigsen, University of Tromsø
- Hilde Tonne, Telenor
- Mats Lundqvist, Chalmers School of Entrepreneurship
- Åshild Grønstad Solheim, PhD student, Simula
- Bjørn Fredrik Nielsen, Research Scientist, Simula

Corporate Management

- Professor Aslak Tveito, Managing Director of Simula Research Laboratory
- Professor Olav Lysne, Director of Basic Research

[1] As of 1 May 2009.

- Dr. Kristin Vinje, Director of Research Education and the Simula School of Research and Innovation
- Cand. Jur., LLM. Ottar Hovind, Director of Research Applications and Director of Administration

Management

- Dr. Joakim Sundnes, Assistant Director of Basic Research
- Professor Are Magnus Bruaset, Assistant Director of the Simula School of Research and Innovation
- Dr. Åsmund Ødegård, Assistant Director of Administration and IT Manager
- Professor Carsten Griwodz, Head of the Networks and Distributed Systems Department
- Professor Hans Petter Langtangen, Head of the Scientific Computing Department, including the Centre of Biomedical Computing
- Dr. Stein Grimstad, Head of the Software Engineering Department
- Dr. Audun Fosselie Hansen, Director of Simula Innovation
- Dr. Christian Tarrou, Director of Kalkulo

B

EXTERNAL CONTRIBUTORS

Bjarne Røsjø
Bjarne Røsjø is a freelance science writer and communications consultant. A biologist from the University of Oslo, Røsjø has worked as a journalist, editor, and advisor in the Norwegian news media and at the Research Council of Norway.

Dana Mackenzie
Dana Mackenzie holds a PhD in Mathematics from Princeton University and works as a freelance mathematics and science writer. Mackenzie has written several books and articles in publications such as Science, Smithsonian Magazine, and New Scientist.

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6,

Christian Hambro
Christian Hambro is an advisor and attorney on matters of administrative and business law, including environmental law, tort law, labour law, and international law. Hambro was director of the Norwegian Research Council from 1994 to 2004.

Anna Godson
Anna Godson holds a degree in Russian and history from Oxford University. Godson works as a freelance translator and also proofreads and edits English texts for a number of clients, including Simula, Telenor, and the Research Council of Norway.

Sverre Christian Jarild
Sverre Christian Jarild is a freelance photographer with a background in press photography. Jarild has also worked with photography in advertising and corporate communications, as well as documentary projects.

Morten Brakestad
Morten Brakestad is a freelance photographer and holds a degree in fine art photography from the Glasgow School of Art. Brakestad has worked as a freelance photographer for newspapers and public relations agencies.

C

COLOUR FIGURES

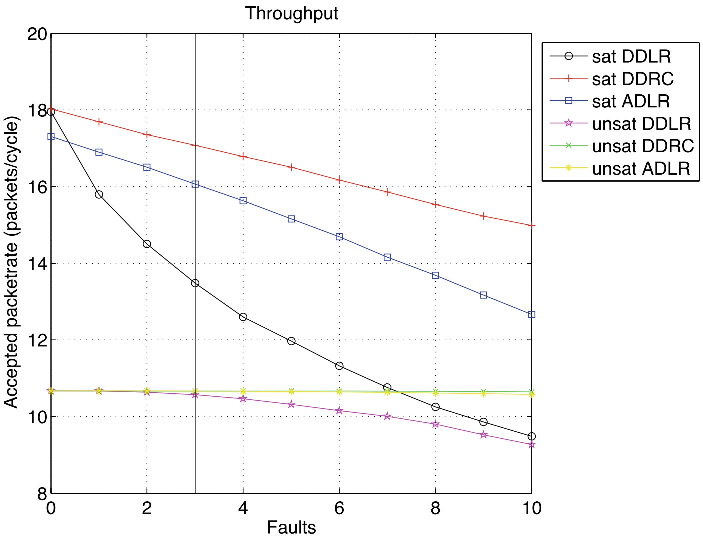
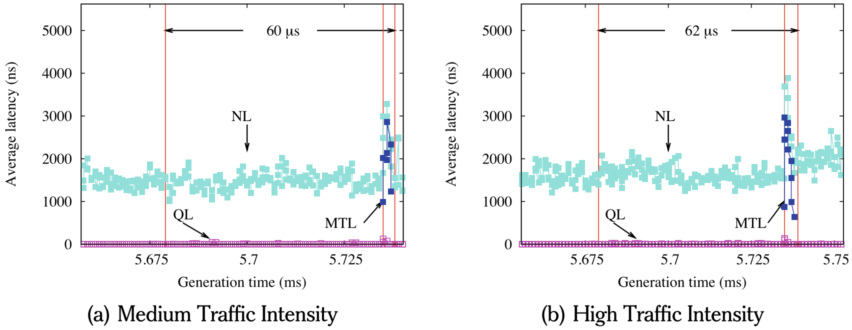


Figure C.1 Throughput in presence of link faults, both under a saturated and unsaturated condition The vertical line marks the transition from guaranteed fault tolerance on the left side to probable fault tolerance on the right side. (This is a colour version of figure 14.5 on page 148.)

A. Tveito et al. (eds.), *Simula Research Laboratory*, DOI 10.1007/978-3-642-01156-6,



(a) Medium Traffic Intensity

(b) High Traffic Intensity

Figure C.2 Average packet latency at various generation times for uniform traffic, where various components of latency are broken down as follows: QL is Queue Latency, NL is Network Latency, and MTL is Maximum Token Latency. The vertical bars represent the start and the end of the reconfiguration respectively. (This is a colour version of figure 14.6 on page 152.)

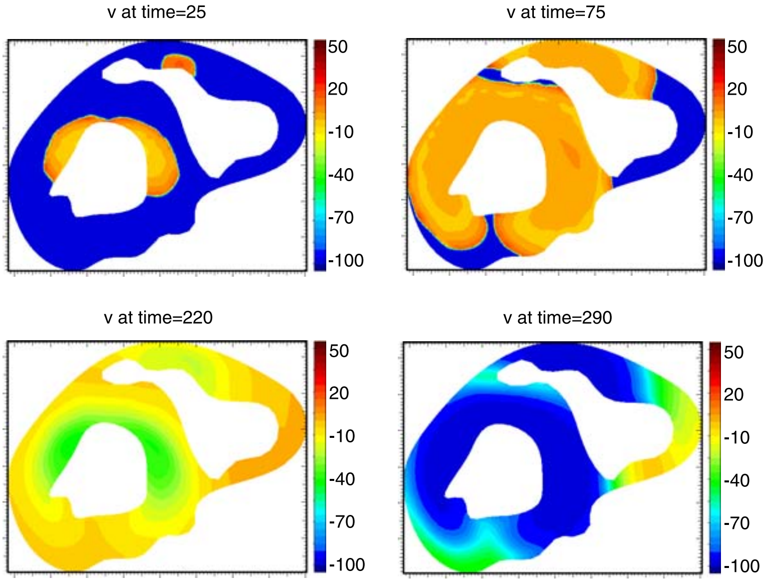


Figure C.3 The transmembrane potential (mV) at four stages. during normal propagation. (This is a colour version of figure 20.2 on page 270.)

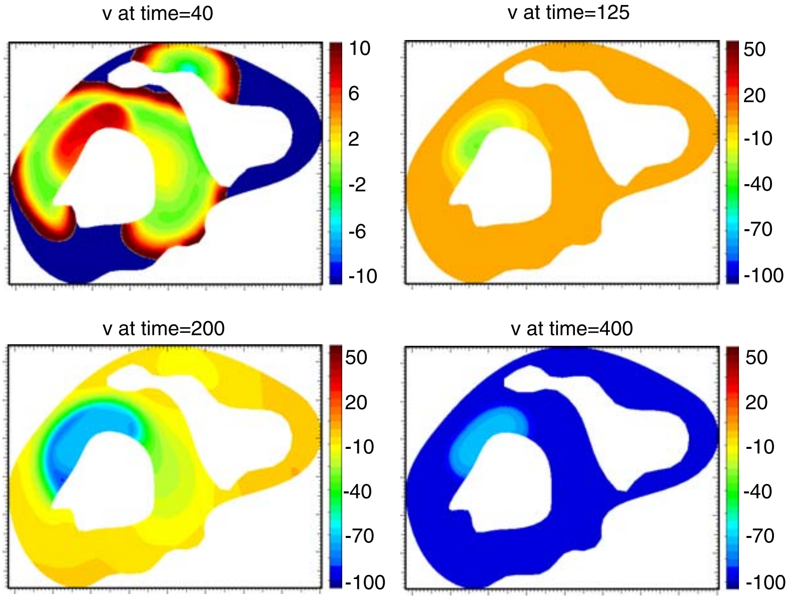


Figure C.4 Four stages during ischemic propagation. (This is a colour version of figure 20.4 on page 272.)

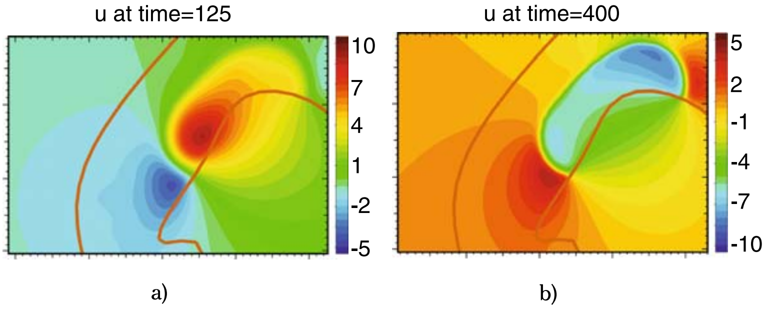


Figure C.5 The extracellular potential around the ischemic area during a) the ST segment and b) the TP segment. The heart boundary is indicated by the solid line. (This is a colour version of figure 20.5 on page 272.)

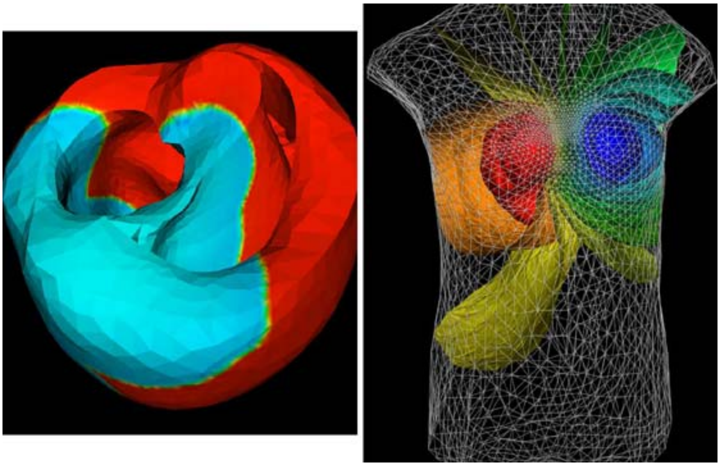


Figure C.6 A snapshot of the transmembrane potential during the early phase of depolarisation. Red tissue is in the resting phase while the blue tissue is depolarised. The right figure shows iso-surfaces of the electrical potential in the torso. (This is a colour version of figure 20.7 on page 273.)

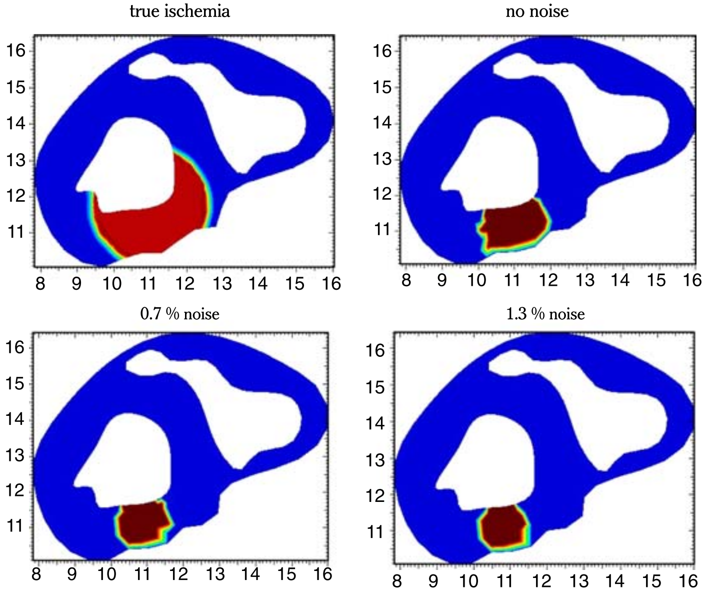


Figure C.7 The location of the true ischemic region and its estimates computed with noise-free and noisy (synthetic) observation data. Note that the position is recovered rather accurately whereas the size of the lesion is underestimated, an issue that should be further explored. (This is a colour version of figure 22.2 on page 299.)

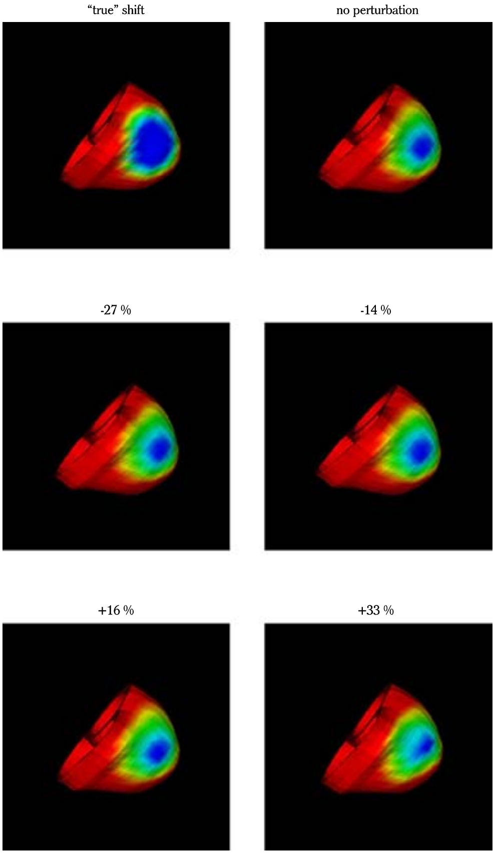


Figure C.8 Results obtained in 3D for a heart in torso model with synthetic observation data. The figures show the “true” and recovered shifts *h* in the transmembrane potential. The numbers above the individual panels quantify the volume perturbations of the heart model used in the inverse solution procedure. The size of the heart is scaled with respect to the size of the panels. (This is a colour version of figure 22.3 on page 304.)

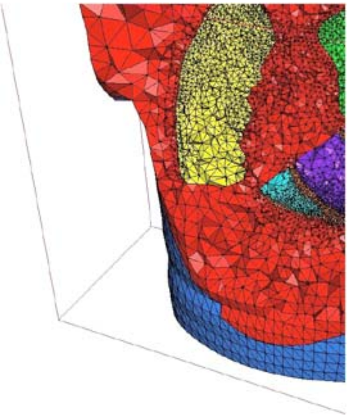


Figure C.9 A tetrahedrisation of the human torso generated from MR images. In addition to the ventricles, our patient-specific models contain lungs. (This is a colour version of figure 22.6 on page 306.)

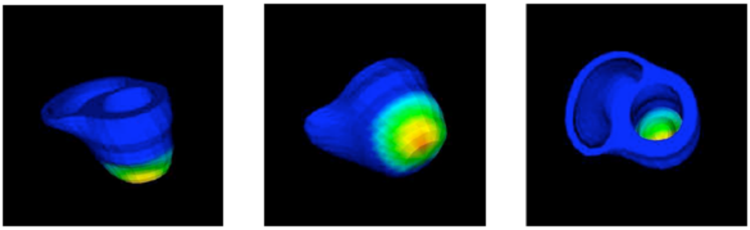


Figure C.10 The ischemic region computed for patient 013 with our inverse ECG model (22.22)-(22.23). The result is consistent with the scintigraphic images shown in figure 22.9. More precisely, the recovered shift h in the transmembrane potential is shown. Blue indicates healthy tissue ($h \approx$ 100mV) and red ischemic tissue ($h \approx 50$mV). (This is a colour version of figure 22.8 on page 307.)

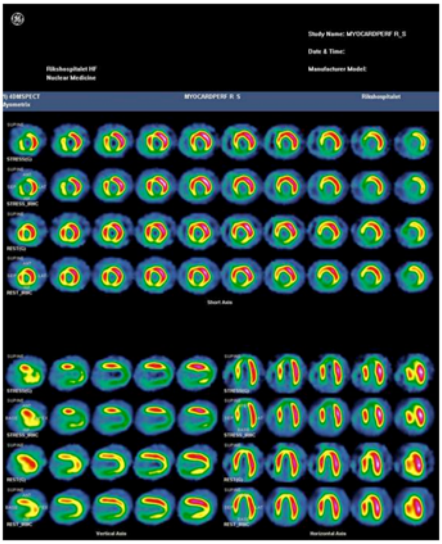


Figure C.11 A 3D scintigram from patient 013 displaying the uptake of the radioisotope technetium-99m in the left heart chamber. The upper two traces show the uptake in horizontal slices during exercise and the next two traces show the uptake during rest. In the lower right panel the same information is depicted on frontal slices and in the left panel on sagittal slices. By comparing the upper and lower two tracings of the sagittal images, it can be seen that there is less uptake in the right and lower parts of the exercise recordings than at rest, corresponding to reversible inferior and apical ischemia. (This is a colour version of figure 22.9 on page 308.)

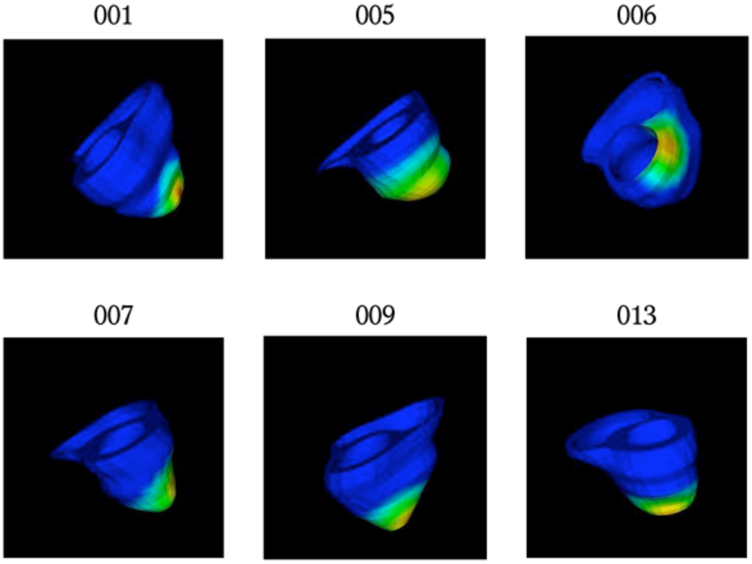


Figure C.12 Inverse solutions computed with the ECG recorded on patient 013 with different geometries. The numbers above each panel specify which geometry was used to produce the result shown. For example, 001 specifies that the geometrical model of patient 001 was employed. More precisely, the estimated shifts h in the transmembrane potential are shown. Blue indicates healthy tissue ($h \approx 100$mV) and red ischemic tissue ($h \approx 50$mV).(This is a colour version of figure 22.10 on page 309.)

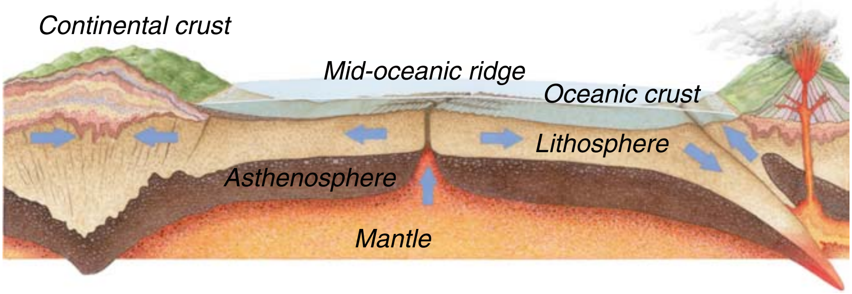
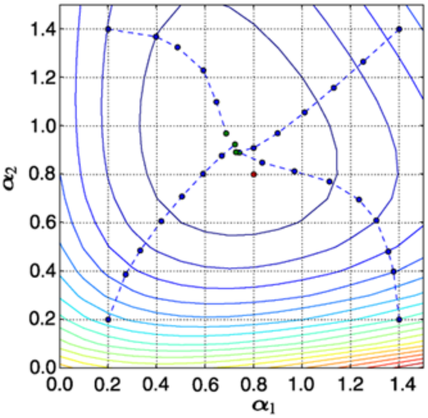


Figure C.13 A vertical section through the Earth exposing the rigid lithosphere and the flowing asthenosphere. The sketch also illustrates the process of lithospheric stretching, by which the lithosphere is thinned to the point of fracturing or breakup, and molten magma arises from the mantle to form a mid-oceanic ridge. The artwork is courtesy of DK Images [55]. (This is a colour version of figure 40.1 on page 557.)



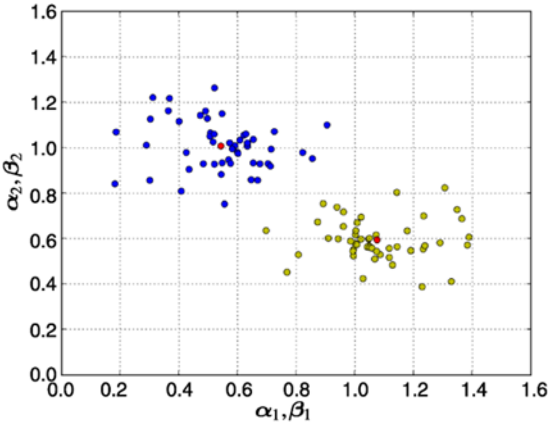


Figure C.14 Top: Using the reference solution with five per cent noise added, the green dots indicate the paths of the Landweber iterates in the α_1-α_2 parameter space for four different starting values. In all cases the iterates converge towards a minimum. This minimum is located in the neighbourhood of the reference parameters $(\alpha_1, \alpha_2) = (0.8, 0.8)$, which are marked with a red dot. Bottom: The four parameters have been calibrated for 50 cases in which the reference solution has been perturbed by 5% random noise. The averages (red dots) approximate the underlying reference parameters $(\alpha_1, \alpha_2) = (0.6, 1.0)$, $(\beta_1, \beta_2) = (1.0, 0.6)$ quite well. The blue dots indicate computed values for α_1 and α_2, whereas the green dots refer to values of β_1 and β_2. The illustrations are courtesy of Hans-Joachim Schroll. (This is a colour version of figure 40.4 on page 568.)

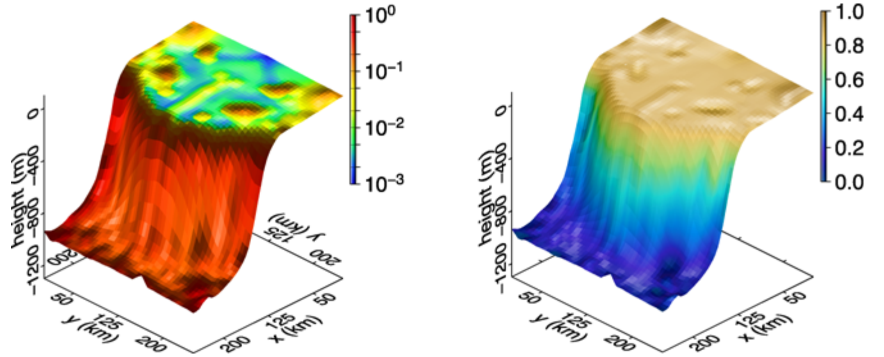


Figure C.15 Expected values (left) and standard deviation (right) of the volume fraction of sand in a Dionisos model for a prograding delta. An even mix of sand, shale and silt is injected into the model at $x = 0$ km, $y = 125$ km and builds a delta out onto the continental shelf. In the mean outcome, sand remains on the contintental shelf and slope, although certain model outcomes allow for significant sand volumes on the lower slope, as is shown on the right. Together these two pictures capture the probability distribution of model outcomes. The illustrations are courtesy of Stuart Clark et al. (This is a colour version of figure 40.7 on page 575.)

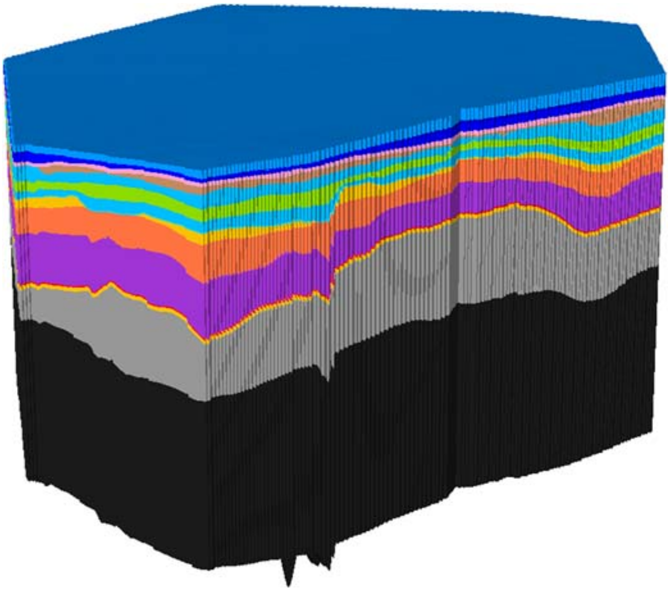
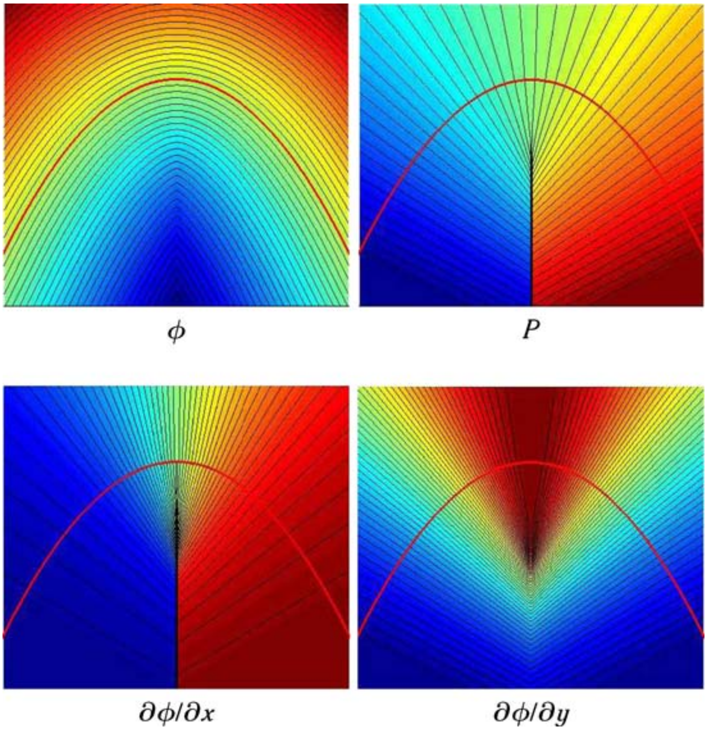


Figure C.16 The Petromod model for the Vøring basin consists of 40 layers with 16 different lithologies. The geometry is embedded in a cubic domain covering 96.8 km × 90.8 km × 34.1 km with a horizontal resolution of 400 meters. The computational problem is defined on a grid with 1.5 million nodes, leading to 7.3 million unknowns. The Vøring data set is courtesy of StatoilHydro. (This is a colour version of figure 40.8 on page 578.)



ϕ P

$\partial\phi/\partial x$ $\partial\phi/\partial y$

Figure C.17 Top, left to right: The computed distance field ϕ and the parameter distribution field P. Bottom, left to right: First and second components of the gradient field $\nabla\phi$ ($\partial\phi/\partial x$ and $\partial\phi/\partial y$). The fields are associated with a simple parabolic curve and are presented in the context of parallel deformation. The illustration is courtesy of Øyvind Hjelle. (This is a colour version of figure 40.11 on page 583.)

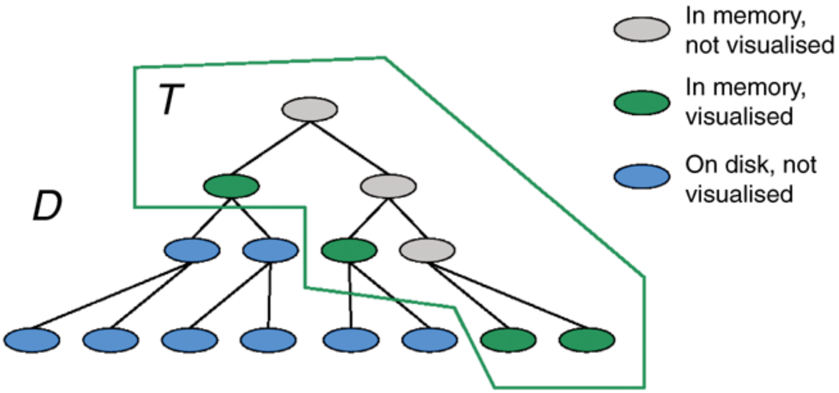


Figure C.18 When a data set is loaded into memory, only parts of the associated persistent tree structure *D* (all nodes, regardless of colour) will be present. This subtree is denoted *T* (grey and green nodes), and only the nodes enabled for visualisation (green nodes) will contribute to the rendering of the scene. Be aware that the grey nodes conceptually will be in memory, even though all or parts of the data values are not yet loaded. (This is a colour version of figure 40.14 on page 587.)

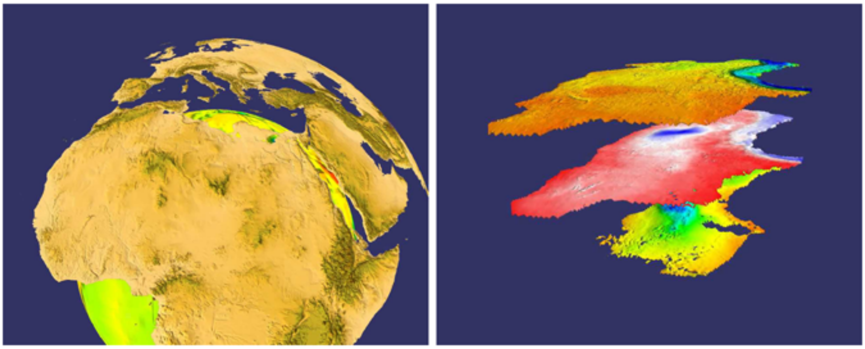


Figure C.19 Left: This two-layer model consists of topography data with the water-covered areas cut away combined with a global surface indicating the bottom of the lower crust. Right: This stack of horizons from the North Sea has only local coverage. The illustrations are courtesy of Trond Vidar Stensby et al. [45]. (This is a colour version of figure 40.15 on page 588.)

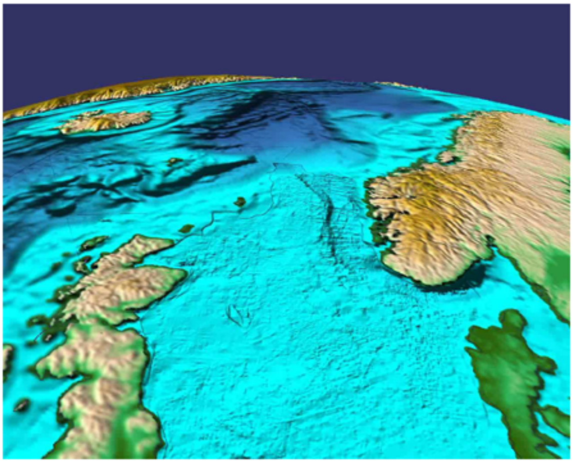


Figure C.20 A high-resolution surface covering a small region in the North Sea has been sewn into a data set for the global topography. The illustration is courtesy of Trond Vidar Stensby et al. (This is a colour version of figure 40.16 on page 589.)

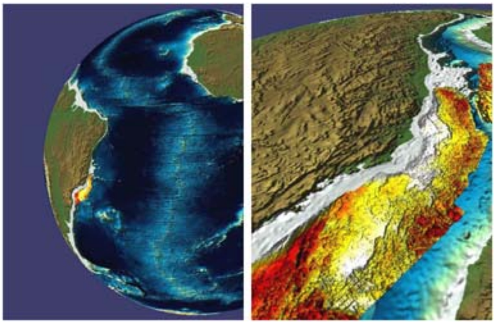


Figure C.21 In the 4DLM, grid-based data can be moved with geological time according to an underlying rotation model. Here, this feature is illustrated by the positions of the tectonic plates for Africa and South America, including regional data sets for base salt layers outside Angola and Brazil. The images show the spatial positions today (left) and 114 million years ago (right). The illustrations are courtesy of Trond Vidar Stensby et al. (This is a colour version of figure 40.17 on page 590.)

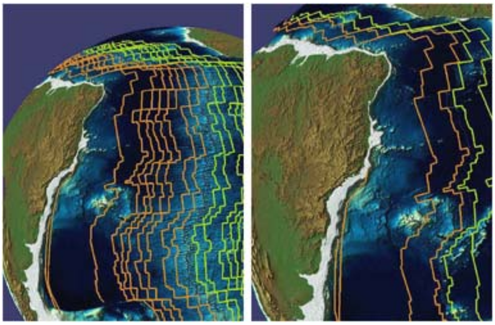


Figure C.22 The 4DLM can rotate both grid-based and vector data, provided that appropriate plate identifiers have been assigned. These images show the locations of isochrons in the South Atlantic today (left) and 60 million years ago (right). The illustrations are courtesy of Trond Vidar Stensby et al. (This is a colour version of figure 40.18 on page 591.)

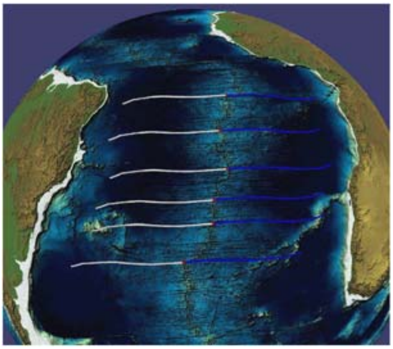


Figure C.23 The flow line shows the time-dependent track of points that originated from a user-selected point at the mid-oceanic ridge in the South Atlantic. This illustration is courtesy of Trond Vidar Stensby et al. (This is a colour version of figure 40.19 on page 591.)