Simulation Foundations, Methods and Applications

Saurabh Mittal
Umut Durak
Tuncer Ören *Editors*

Guide to Simulation-Based Disciplines

Advancing Our Computational Future

Springer

Simulation Foundations, Methods and Applications

More information about this series at http://www.springer.com/series/10128

Saurabh Mittal · Umut Durak · Tuncer Ören
Editors

Guide to Simulation-Based Disciplines

Advancing Our Computational Future

Springer

Editors
Saurabh Mittal
MITRE Corporation
McLean, VA
USA

Umut Durak
German Aerospace Center (DLR)
Braunschweig
Germany

Tuncer Ören
University of Ottawa
Ottawa, ON
Canada

ISSN 2195-2817 ISSN 2195-2825 (electronic)
Simulation Foundations, Methods and Applications
ISBN 978-3-319-61263-8 ISBN 978-3-319-61264-5 (eBook)
DOI 10.1007/978-3-319-61264-5

Library of Congress Control Number: 2017943822

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To the pioneers of simulation,
who used models and computation to generate new knowledge that opened new frontiers

To Vandana and Ekaansh and Reyaansh,
My wife and my sons, who create the life I love, every single moment
To the Almighty,
who guides and inspires in interesting ways

Saurabh Mittal

To Rabia,
My beloved companion in my never-ending story

Umut Durak

To Füsun,
My wife and lifelong friend

Tuncer Ören

Foreword

Nothing like this book is on the market today. Of course, all books make a similar claim but there really is no book that offers to guide readers toward anchoring their disciplines on modeling and simulation (M&S)—in fact, toward a far-reaching concept of *simulation-based disciplines*. Why is such a guide needed? Most disciplines employ models, some to a larger extent than others. Further, today almost all disciplines employ computers and very often this includes methods that are referred to as simulation. However, it is less appreciated, that we are witnessing the development of M&S as a discipline itself. Indeed, M&S is contributing to the enhancement of many other disciplines and its perspectives and methods are being absorbed into full-fledged practice. In this light, the editors of this book attest that "simulation is mature enough to provide a solid basis for advancement in many disciplines, from life sciences to engineering, from architecture to arts to social sciences". So the advance of your discipline, and indeed, the advance of your own career in it, warrant checking out both the general chapters that bring you up to date on the current state of M&S as a discipline, as well as the chapters that discuss the benefits that M&S is bringing to a selected array of disciplines.

This one-of-its-kind book is concerned with how computational methods embodied in M&S are used and how they push forward science and technology. As is befitting of a book that spans disciplines, its editors and authors bring a diverse range of topical, national and global perspectives. Each of these perspectives throws somewhat different light on the common theme that computers and computation take central roles in the accelerating advance of human knowledge. On this critical journey, modeling and its knowledge generation vehicle, simulation, will increasingly drive the advance of disciplines.

Potomac, MD, USA Bernard P. Zeigler

Preface

Motivation

We live in the information age and every technology in today's modern world is based on information technology. Scientific theories—introduced in the last five decades—are realized with today's scalable computational infrastructure effectively characterized by Moore's Law. Modeling and Simulation (M&S), along with Big Data technologies, is at the forefront of such exploration and investigation. Furthermore, simulation has a unique characteristic: Dynamic models associated with diverse experimental conditions and scenarios (sometimes in extreme and even at adverse conditions) have the power of generating new knowledge.

M&S is often taken as a single activity but in practice and engineering, are separate activities. While modeling has been adequately embraced by various disciplines, leading to many modeling techniques, many times the modeling activity is not supported by simulation activity. On the other hand, a simulation-based approach subsumes modeling as an inherent activity. As editors, we believe while modeling helps in bringing a common understanding across all stakeholders (e.g., scientists, engineers, practitioners), it is usually through simulation that a model's correctness is evaluated. A validated model must be amenable to simulation. A *model* represents a real-world phenomenon using abstractions. A properly validated and verified and properly computerized model lends itself to experimentation. Experimentation with a dynamic model as well as gaining experience based on a dynamic model lies within the domain of simulation-based approaches.

This book has several examples from diverse disciplines demonstrating simulation maturity to provide a solid basis for advancement in many disciplines; from life sciences to engineering; from architecture to arts to social sciences. As a sign of the times, we are truly at the crossroads where M&S is becoming a "discipline". This book emphasizes the fact that simulation may enhance the power of many disciplines. Not only is M&S benefiting disciplines like sociology, which has largely been insulated from such experimentation, but it is also undoubtedly used in every aspect of life, whether transportation, finance, economics, biology, and so forth.

The book's emphasis is on highlighting the state-of-the-art simulation in the modern era and how simulation-based approaches across multiple disciplines advance the very discipline itself.

Overview

The book is organized in Background, Engineering and Architecture, Natural Sciences, Social Sciences and Management, and Learning, Education and Training sections. We appreciate very much our eminent colleagues who accepted our invitations to contribute to this volume.

The Background section endeavors to provide a comprehensive simulation view in two chapters. In Chap. 1, the editors, Tuncer Ören, Saurabh Mittal and Umut Durak, try to establish a base for the book by elaborating simulation evolution and highlighting simulation as vital infrastructure for many disciplines. Ernest H. Page presents in Chap. 2 the simulation technology landscape in academia, industrial and government sectors throughout various scientific and engineering disciplines.

The Engineering and Architecture section is composed of six chapters each drawing attention to a certain technical domain. In Chap. 3, Melih Çakmakcı, Güllü Kızıltaş Şendur, and Umut Durak address the role of simulation in engineering design. Andreas Tolk, Christopher G. Glazner, and Robert Pitsko promote simulation in Chap. 4 as an evolution of model-based systems engineering toward an integrated discipline within systems engineering. Chapter 5 Simulation-Based Cyber-Physical Systems and Internet of Things by Bo Hu Li, Lin Zhang, Tan Li, Ting Yu Lin, and Jin Cui explains the relation of simulation with these emerging fields of technical systems. In Chap. 6, Saurabh Mittal and Jose Luis Risco Martin accentuate complex and adaptive systems and introduce required simulation infrastructure for their design. Following that, Chap. 7 is from Oryal Tanir, where he presents simulation as a means of understanding the impact of a new complex solution on the business and information technology process in a software design life cycle. The last chapter of this section is by Rhys Goldstein and Azam Khan. They introduce the emerging role of simulation in designing model compelling, functional, sustainable, and cost-effective buildings in Chap. 8.

The Natural Sciences section includes two chapters. First in Chap. 9, Levent Yilmaz conducts a comprehensive discussion on the position of simulation in relation to scientific method towards simulation-based science. Then in Chap. 10 Hannes Prescher, Allan H. Hamilton, and Jerzy Rozenblit introduce the contribution of simulation to health-care as well as health education and training.

The Social Sciences and Management section contains two chapters. It starts with Chap. 11 where David C. Earnest and Erika Frydenlund introduce simulation evolution in social sciences and discuss its position in research. In Chap. 12 Greg Zacharewicz, Amir Pirayesh-Neghab, Marco Seregni, Yves Ducq, and Guy

Doumeingts present simulation of service systems for simulation-based enterprise management.

The last two chapters establish the Learning, Education and Training section. In Chap. 13, Tuncer Ören, Charles Turnitsa, Saurabh Mittal and Saikou Diallo present the role of simulation in learning and education. Finally, the last chapter by Agostino Bruzzone and Marina Massei discusses the role of simulation in military training.

Invitation

The notable contribution of this book is providing a comprehensive collection of chapters from diverse disciplines with the unique characteristics of simulation. Authors explore and elaborate the position of simulation within their domain and note its impact for the advancements of their disciplines. We invite you to a journey about simulation through various disciplines and anticipate that such a synergistic approach will provide you an overview of the role of simulation as we are advancing towards a computational future in the twenty-first century: a computational future to be enhanced and empowered by simulation-based approaches.

Herndon, VA, USA — Saurabh Mittal[1]
Braunschweig, Germany — Umut Durak
Ottawa, ON, Canada — Tuncer Ören
February 2017

[1]The authors affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved for public release: Case: 17-0904.

Contents

Editors and Contributors

About the Editors

Saurabh Mittal is Lead Systems Engineer/Scientist at MITRE, founder and President of Dunip Technologies, LLC and Vice President at Memberships for Society of Computer Simulation (SCS) International. He is also currently serving as Associate Editor-in-Chief for Transactions of SCS and Editor-in-Chief of Enterprise Architecture Body of Knowledge (EABOK) Consortium. Previously, he was a full-time scientist and architect at National Renewable Energy Laboratory, Department of Energy at Golden, Colorado, where he contributed to complex energy systems co-simulation environments. He also worked at L3 Link Simulation & Training as a contractor to US Air Force Research Lab at Wright-Patterson Air Force Base, Ohio, where he integrated artificial agents and various cognitive architectures in Live, Virtual and Constructive (LVC) environments using formal systems theoretical model-based engineering approaches. He was Research Assistant Professor at the Department of Electrical and Computer Engineering at the University of Arizona. Dr. Mittal served as General Chair of Springsim’17 and SummerSim’15, Vice General Chair for SpringSim’16 and SummerSim’14, and Program Chair for SpringSim’15. He is a recipient of US Department of Defense (DoD) highest civilian contraction recognition: Golden Eagle award and SCS’s Outstanding Service award. He has coauthored nearly 60 articles in various international conferences and journals, including a book titled “Netcentric System of Systems Engineering with DEVS Unified Process”, CRC Press in 2013 that serve the areas of executable architectures, service-oriented distributed simulation, System of Systems engineering using Department of Defense Architecture Framework (DoDAF), multiplatform modeling, intelligence-based, complex, adaptive and autonomous systems, and large-scale M&S integration and interoperability.

Umut Durak is a research scientist in the Institute of Flight Systems at the German Aerospace Center. He is also Adjunct Lecturer in the Department of Informatics at Clausthal University of Technology. His research interests include model-based simulation engineering, simulation-based systems engineering and ontologies in simulation. He received his BS, MS and Ph.D. degrees in Mechanical Engineering from Middle East Technical University (METU) in Turkey. He is a member of the Society for Computer Simulation International (SCS) and Arbeitsgemeinschaft Simulation (ASIM) and a senior member of the American Institute of Aeronautics and Astronautics (AIAA). And, he is a part of AIAA Modeling and Simulation Technical Committee and editorial teams of the Simulation: Transactions of the Society for Modeling and Simulation International; and the International Journal of Modeling, Simulation, and Scientific Computing.

Tuncer Ören is Emeritus Professor of Computer Science at the University of Ottawa, Canada. He received his Ph.D. in Systems Engineering from the University of Arizona. He has been active in simulation since 1965. His **research interests** include: (1) advancing modeling and simulation methodologies; (2) agent-directed simulation (full synergy of simulation and agents); (3) agents for cognitive and emotive simulations; (4) reliability, quality assurance, failure avoidance, and ethics; as well as (5) body of knowledge and (6) multilingual terminology of modeling and simulation. Dr. Ören has over 500 **publications**, including 40 books and proceedings. He has contributed to over 500 **conferences** and seminars (about half as invited/honorary) in about 40 countries. He was inducted to SCS Modeling and Simulation Hall of Fame—Lifetime Achievement **Award**. He is also a distinguished lecturer, an SCS fellow, and AVP for ethics. He is recognized, by IBM Canada, as a pioneer of computing in Canada. A book about him was edited in 2015 by Prof. L. Yilmaz: Concepts and Methodologies for Modeling and Simulation: A Tribute to Tuncer Ören, Springer. **Other distinctions** include invitations from United Nations and plaques and certificates of appreciation from organizations including ACM, Atomic Energy of Canada, and NATO.

Contributors

Agostino G. Bruzzone Simulation Team, Savona, Italy

Melih Cakmakci Department of Mechanical Engineering, Bilkent University, Ankara, Turkey

Jin Cui School of Automatic Science and Electrical Engineering, Beihang University (Beijing University of Aeronautics and Astronautics), Beijing, China

Saikou Y. Diallo Old Dominion University, Norfolk, USA

Guy Doumeingts Laboratory IMS CNRS Production Engineering Team, Talence Cedex, France; INTEROP-VLab, Brussels, Belgium

Yves Ducq Laboratory IMS CNRS Production Engineering Team, Talence Cedex, France

Umut Durak Institute of Flight Systems, German Aerospace Center (DLR), Braunschweig, Germany

David C. Earnest Department of Political Science & Geography, Old Dominion University, Norfolk, USA

Erika Frydenlund Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA, USA

Christopher G. Glazner The MITRE Corporation, McLean, VA, USA

Rhys Goldstein Autodesk Research, Toronto, ON, Canada

Allan J. Hamilton Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA

Azam Khan Autodesk Research, Toronto, ON, Canada

Gullu Kiziltas Sendur Mechatronics Engineering Program, Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

Bo Hu Li School of Automatic Science and Electrical Engineering, Beihang University (Beijing University of Aeronautics and Astronautics), Beijing, China; Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing, China; State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing, China

Tan Li Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing, China; State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing, China

Ting Yu Lin Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing, China; State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing, China

Marina Massei Simulation Team, Savona, Italy

Saurabh Mittal The MITRE Corporation, McLean, USA

Tuncer Ören School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada

Ernest H. Page Modeling, Simulation, Experimentation and Analytics Technical Center, The MITRE Corporation, Mclean, USA

Amir Pirayesh-Neghab INTEROP-VLab, Brussels, Belgium

Robert Pitsko The MITRE Corporation, McLean, VA, USA

Hannes Prescher Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA

José L. Risco-Martín Complutense University of Madrid, Madrid, Spain

Jerzy W. Rozenblit Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA

Marco Seregni Department of Management, Economics and Industrial Engineering, Politecnico Di Milano, Milan, Italy

Oryal Tanir Office of the CIO, McGill University, Montreal, Canada

Andreas Tolk The MITRE Corporation, Hampton, VA, USA

Charles Turnitsa Georgia Tech Research Institute, Atlanta, USA

Levent Yilmaz Computer Science and Software Engineering, Auburn University, Auburn, AL, USA

Gregory Zacharewicz Laboratory IMS CNRS Production Engineering Team, Talence Cedex, France

Lin Zhang School of Automatic Science and Electrical Engineering, Beihang University (Beijing University of Aeronautics and Astronautics), Beijing, China

Acronyms

AFAMS US	Air Force Agency for Modeling and Simulation
AFIT	US Air Force Institute of Technology
AFRL	US Air Force Research Laboratory
AGI	Analytical Graphics, Inc
AMSO	US Army Model and Simulation Office
ANL	Argonne National Laboratory
ARL	US Army Research Laboratory
ASIM	Arbeitsgemeinschaft Simulation
DSTL	Defence Science and Technology Laboratory
iDSC	interim Defence Simulation Centre
FIU	Florida International University
GMU	George Mason University
LANL	Los Alamos National Laboratory
LLNL	Lawrence Livermore National Laboratory
NAWC TSD	Naval Air Warfare Center Training Systems Division
NCSU	North Carolina State University
NDIA	National Defense Industrial Association
NPS	Naval Postgraduate School
NTSA	National Training and Simulation Association
ORNL	Oak Ridge National Laboratory
PMTRASYS	US Marine Corps Systems Command Program Manager for Training Systems
SAP	Systems, Applications & Products in Data Processing
SIMAF	Simulation and Analysis Facility
SFI	Santa Fe Institute
SNT	Scalable Networking Technologies
TRADOC	US Army Training and Doctrine Command
UCF/IST	University of Central Florida Institute for Simulation and Training
USC/ICT	University of Southern California Institute for Creative Technologies

Part I
Background

Chapter 1
The Evolution of Simulation and Its Contribution to Many Disciplines

Tuncer Ören, Saurabh Mittal and Umut Durak

We often fail to realize how little we know about a thing until we attempt to simulate it on a computer.
Donald E. Knuth
From: The Art of Computer Programming, Volume 1—Fundamental Algorithms, 1968

Abstract The aims of this chapter are: (1) To provide a comprehensive view of the stages of the evolution of simulation. (2) To emphasize the phenomenal developments in many aspects of simulation which made it an important and even a vital infrastructure for many disciplines. (3) To underline the fact that the transition from "model-based" paradigm to "simulation-based" paradigm may be beneficial for many disciplines. In Sect. 1.2, references for a systematic collection and a critical review of about 100 definitions of simulation as well as a comprehensive and integrative definition of simulation are given. In Sect. 1.3, the reasons simulation is used are clarified. These reasons make simulation very useful for many disciplines. In Sect. 1.4, nine aspects of the evolution of simulation are clarified including simulation-based disciplines. In Sect. 1.5, many disciplines for which simulation-based paradigm would make them much more powerful and efficient are elaborated.

The authors affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved for public release: Case: SM_001

T. Ören (✉)
University of Ottawa, Ottawa, Canada
e-mail: oren@eecs.uOttawa.ca

S. Mittal
The MITRE Corporation, Virginia, USA
e-mail: smittal@mitre.org

U. Durak
German Aerospace Center (DLR), Braunschweig, Germany
e-mail: umut.durak@dlr.de

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_1

Keywords Agent Simulation · Agent-based simulation · Agent-directed simulation · Agent-monitored simulation · Computerized simulation · Evolution of simulation · Experience · Experiment · Formal simulation · Non-computerized simulation · Reasons to use simulation · Similitude · Simulation systems engineering · Simulation-as-a-service · Simulation-based · Simulation-based discipline · Soft simulation · Terms related with similitude · Thought experiment

1.1 Introduction

The term "simulation", derived from Latin *similis* "like" has existed in the English language since the mid-fourteenth century. The concept of similarity which is also related with Latin *similis*, is very rich. The Appendix gives—in 14 groups—a list of terms associated with similarity. "Starting with the etymology of simulation, *simulare* is a Latin verb originally denoting the act of making one thing be similar to another. From this fundamental meaning derive connotations such as *to pretend*, *to falsify*, *to feign*, and *to make believe*. As Heidegger noted in two of his 1957 lectures (published posthumously under the title Identity and Difference), the capability to recognize when things are the same and when they are different from one another is a fundamental precondition for the construction of any ontology" (Gualeni). The term simulation still represents the original concepts for which it was coined (ED—simulation), in addition to many powerful contemporary technical concepts that are pointed out in this chapter.

Since simulation has many aspects, meanings, and associated connotations, it is appropriate to have a comprehensive and integrative view. For this reason, we offer a comprehensive view of simulation before the presentation of the stages of the evolution of simulation. We emphasize the fact that the phenomenal developments in many aspects of simulation have made it an important and even a vital infrastructure for many disciplines. Finally, we share our view that transition from "model-based" paradigm to "simulation-based" paradigm may be beneficial for many disciplines.

In Sect. 1.2, references for a systematic collection and a critical review of about 100 definitions of simulation as well as a comprehensive and integrative definition of simulation are given. In Sect. 1.3, the reasons simulation is used are revised. These reasons make simulation very useful for many disciplines. In Sect. 1.4, nine aspects of the evolution of simulation are clarified including simulation-based disciplines. In Sect. 1.5, many disciplines for which simulation-based paradigm would make them much more powerful and efficient are elaborated.

1.2 Definition(s) of Simulation

Some terms were used in English even before the technical aspects were attributed to them. An example is the term "computer" which originally denoted, in the middle of the seventeenth century, a human, "one who computes but now the term almost universally refers to automated electronic machinery" (Swaine et al. 2017). With this new meaning, the original connotation was dropped. Similarly, the term "simulation" has been used in English, since mid-fourteenth century. Contrary to the term "computer", the term "simulation" retained its original meanings, in addition to its technical meanings. Hence, elaboration of the concept of similarity may cast light even on simulation.

Similarity has basically three meanings. All three aspects cover a plethora of other meanings.

1. Existence of common characteristics between two entities. This first meaning covers four basic shades and the shades in between:

   1.1 Two entities are (almost) identical
   1.2 Two entities do not have any common characteristic
   1.3 The (in)ability of perceiving the similarities
   1.4 Hiding the similarities

2. Pretention. This meaning covers the following:

   2.1 Give the impression of something to (re-)create a situation (e.g., fiction which can create or recreate a situation.)
   2.2 Give the impression of something for faking (e.g., halo effect)

3. Imitation. Depending on the goal, "imitation" and "similarity" can also denote counterfeit.

   3.1 Replication of another person's behavior (emulation, parroting, counterfeit appearance: mockery)
   3.2 Replication of another object's characteristics (e.g., simulated leather, simulated pearl)
   3.3 Copying, reproduction, replica (e.g., imitation diamond).

To show the richness of the concept of similarity—some of which are applicable to several aspects of simulation—the Appendix lists several terms related with the concept of similarity. The terms are grouped under the following 14 categories:

- simulation concept, model;
- analogy, imitation, behavioral similarity, functional similarity, similarity in mathematics, similarity in linguistics, similarity in literature, similarity in art, to be similar; and
- indistinguishableness, disguise similitude under a false appearance, and non-similarity.

This fact and the fact that technical meaning of “simulation” embraces several concepts create some confusion. For this reason, a clarification may be useful. A systematic collection of about 100 definitions of simulation (Ören 2011a) shows that some of the definitions are about some aspects or types of simulation and are not comprehensive and integrative. A critical review of these definitions is also available (Ören 2011b). A concise, albeit comprehensive and integrative, definition of simulation follows:

> Simulation is performing goal-directed experimentation or gaining experience under controlled conditions by using dynamic models; where a dynamic model denotes a model for which behavior and/or structure is variable on a time base.

1.3 Reasons to Use Simulation

Simulation is used for many reasons. These reasons are also the essence of the power of simulation that can be used by any other discipline. As early as 1970s, Karplus (1977) presented a spectrum of problem types to explain the reasons to use simulation. The spectrum—which ranges from arousing public opinion to product design—covers gaining insight, testing theories, experimentation with control strategies, and prediction for both action and performance. These reasons perfectly overlap with the experimentation and experience aspects of simulation. Arousing public opinion is a motivating task from social sciences such as political science or economics, gaining insights, and testing theories are the typical tasks from life sciences or physical science that employ simulation. Experimentation with control strategies, prediction for both action and performance, and product design are the conventional tasks of engineering that typically require the use of simulation.

Within the *Experimentation aspect*, simulation is used for:

- behavior prediction and performance analysis;
- analysis of alternatives;
- sensitivity analysis;
- engineering design;
- virtual prototyping;
- planning;
- acquisition; and
- proof of concept.

Within the *Experience aspect*, simulation (sometimes as virtual reality (VR), and augmented reality (AR)) is used for:

- training to enhance one of the three types of skills, i.e., motor skills (virtual simulation), decision-making skills (constructive simulation), and operational skills (live simulation);
- entertainment (simulation games); or
- shared knowledge and emotions, as in art (Dewey 1934) and literature. For the last category, visual renderings are theatre, movie, and TV.

As seen in Table 1.1, there is also a set of task characteristics that requires the use of simulation (Ören 2005). In most of the design problems, the real system does not exist. In some cases, even if it exists, it is not accessible for experimentation such as in space exploration problems. The dynamics and the response of the system also play an important role in determining what task could be supported. Both too slow dynamics, such as economic studies, and too fast dynamics, such as particle physics, make the observations almost impossible. The safety considerations may also, sometimes, dictate a simulation-based experimentation. Conducting test for aircrafts out of their flight envelopes is an example of that. In other cases, experimentation may not be acceptable by populace, such as experimentation with education systems. The cost of experimentation is also another driver for simulation use. Conclusively, it is more convenient to use simulation when experimentation conditions cannot be fulfilled physically, for example, the range of the system variables cannot be controlled or are unachievable. These characteristics also persuade the use of simulation.

Another perspective to use simulation (for teaching) is given by JeLSIM:

"Simulations provide:

- **Exploration**—learners can explore domains that would otherwise be too time-consuming, expensive or dangerous.

Table 1.1 Reasons to use simulation [adapted from Ören (2005)]

Aspect	Reason
Real system	The real system does not exist (as in design problems)
	The real system is not accessible for experiments (as in deep sea or space exploration problems)
	The dynamics and response of the real system is too slow or too fast for observation (e.g., geological studies, economic studies; particle physics)
Experiment	The experiment is dangerous (e.g., extreme cases in pilot training, study of a failure in a levee)
	The experiments are unacceptable by public (experimenting cases where public would be affected directly, e.g., experimenting different public transportation policies)
	The experimenting with real system is not cost-effective (e.g., use of physical prototypes versus computerized simulations)
	The proper conditions for the experiment cannot be fulfilled
Variables	The variables of the system (as opposed to simulation) cannot be controlled
	Process variables cannot be measured on the real system
	Measurements of the variables of the real system would be noisy

- **Focus**—they facilitate the removal of complexity and detail from a model, focusing only on the aspects of the model that are most relevant to the learning.
- **Visualization**—they make it easier to visualize dynamic or complex behavior.
- **Motivation**—they motivate by providing context and engagement, encourage active involvement, and arouse interest.
- **Control**—learners can control timing and detail, they can explore and experiment, hypothesize and test.
- **Practice**—to address misconceptions and allow learners to learn from their mistakes."

1.4 Aspects of the Evolution of Simulation

The concept of simulation has evolved over the ages and with the advent of computer technologies, it gained importance in every sphere of study. The capabilities provided in experimentation and experience aspects have evolved with the increasing computing power. The realization of Moore's Law has provided the needed catalyst for simulation technology that is now pervasive.

The evolution of simulation can be easily understood in two broad categories

- Noncomputerized simulation
- Computerized simulation.

Evolution of simulation can be studied within nine aspects that are depicted in Fig. 1.1. The first aspect is noncomputerized simulation which was the answer to the needs of experimentation and gaining experience, albeit not on the real system. Addition of "computation" results in computerized simulation which is also called computational simulation, computer simulation, in silico simulation, or in short, simulation. Since the term "computer simulation" also means simulation of computer systems, when the term "computer simulation" is used, the meaning is understood based on the context.

Similarly, each aspect represents the influence of some additional feature. The identified nine aspects of the evolution of simulation are: noncomputerized simulation, computerized simulation, formal simulation, AI-directed simulation, agent-directed simulation, soft simulation, simulation systems engineering, simulation-based disciplines, and simulation-as-a-service. These aspects of the evolution of simulation should not be interpreted as levels of simulation even though sometimes they may appear to be. For example, computerized simulation is a more advanced version of noncomputerized simulation. However, some aspects, such as formal simulation and agent-directed simulation may exist simultaneously.

1.4.1 Noncomputerized Simulation

Simulation, in the sense of pretending (to make believe, to claim, represent, or assert falsely), has been used since a long time in relation with both experimentation and experience.

	Aspects of simulation	Additional feature
9.	Simulation-as-a-service	Service
8.	Simulation-based disciplines	Infrastructure
7.	Simulation systems engineering	Systems Engineering
6.	Soft simulation	Soft computing
5.	Agent-directed simulation	Autonomy
4.	AI-directed simulation	Intelligence
3.	Formal simulation	System theories
2.	Computerized simulation	Computation
1.	Non-computerized simulation	

Fig. 1.1 Aspects of simulation

Experimentation done by pure thinking is called *thought experiment* (also, conceptual experiment or *Gedankenexperiment*). Thought experiments have been used mostly in ethics, philosophy, and physics. Some examples are prisoner's dilemma and trolley problem (Brown et al. 2014). Physical aids, such as *scale models*, were also used for simulation done for experimentation purposes. Another possibility has been simulation of the real system under controlled experimental conditions, such as wheel and tire simulators.

Experience aspect of covers three major areas for training, entertainment, and for shared knowledge and feelings. *Training* is done to enhance skills. *Role playing* has been a use of simulation to gain experience under controlled conditions for training to enhance necessary skills. For example, training of a junior sales representative while a senior representative may act as a prospective customer. Some historic examples to the use of physical aids to gain experience for training purposes include sand-box simulation, used for military training and mechanical simulators. Simulation for *entertainment* purposes has a wide application. Any game where a pretention about reality is concerned falls in this category. From a philosophical perspective, the book by John Dewey, titled: "Art as experience" is a classical one

Table 1.2 Non-computerized simulation

Goal	Pure thinking	Physical aids
Experimentation	Thought experiments	Simulation with scale models
Experience		
– for training (to enhance skills)	Role playing	Sand-box simulation Mechanical simulators
– for entertainment	Games	
– for shared knowledge and emotions	Art (literature, painting, sculpture)	Visual renderings (theatre, movie, TV)

(Dewey 1934). Several references exist about "Literature as experience" (Wallace and Breen 1959; Howe 1979). Any work of literature is a created reality where the reader faces a make-believe situation. The source can be pure fiction or reality. A reference for the relationship of simulation and reality is given by Ören (2010). In visual renderings of works of literature, such as in theater, movie, or TV, physical aids are also used as part of the décor and/or environment. It is interesting to note that, in literature, the author creates a work of art based on fictional (imagined) or existing reality. In plastic arts, such as sculpture or painting, the artist creates her work of art based on "reality" that is called a "model" (Table 1.2).

1.4.2 Computerized Simulation

Computerized simulation, or computational simulation is also called computer simulation to make it shorter. However, the term "computer simulation" also means simulation of computer systems, like "traffic simulation" which means simulation of traffic. The distinction of these two meanings is done based on the context within which the term is used. In this section, the term "computerized simulation" is used to mean that some of the activities of simulation are performed by computers. As indicated in Table 1.3, in the early days, computers were used only for behavior generation (Ören 1982a). Some of the types of computer-aided model processing were discussed as early as 1980s (Ören 1983). Table 1.4 outlines influence of types of computers, to the advancement of simulation: some of the types of computers such as wearable, implantable, and quantum computers, are emerging.

Table 1.3 Computerized simulation (computer simulation)—role of computers

Role of computers	Implication for simulation
– behavior generation	Initially computers are used only for behavior generation
– also, problem specification	Computer-aided simulation environments

Table 1.4 Computerized simulation (computer simulation)

Type of computer	Type of simulation
Analog computer	Analog simulation
Hybrid computer	Hybrid simulation
Digital computer	Digital simulation
Batch processing computer	Noninteractive simulation
Interactive computer	Interactive (Online) simulation
Distributed computing	Distributed simulation
Cloud computing	Cloud simulation
Wearable computer	Wearable simulation
Implantable computer	Implantable simulation
High-performance computer	High-performance simulation
Terascale computer	Terascale simulation
Petascale computer	Petascale simulation
Exascale computer	Exascale simulation
Quantum computer	Quantum simulation

1.4.3 Formal Simulation

In the early days of computerized simulation, craftsmanship of simulation was promoted. See for example, Tocher (1963). Importance and a way to separate models from experimentation and other aspects of behavior generation were documented for the first time in 1979 (Ören and Zeigler 1979). A first conference—supported by NATO—on simulation and model-based methodologies was held early in the 1980s (Ören et al. 1984; Ören 1984).

Separation of concerns—especially model—in simulation, marked the beginning of the era of model-based simulation. Later, model-based other activities, such as model-based systems engineering (Wymore 1993) and model-based software engineering were also established. System theoretic bases of simulation were the essence of system theory-based simulation. A first model specification language (as opposed to simulation programming language) based on a system theory was GEST (General System Theory implementor (Ören 1971). The history of system theoretic bases of simulation was elaborated on by Ören and Zeigler (2012). Indeed, Zeigler's DEVS (Discrete Event System Specification) (Zeigler 1976)—as an extension of Moore machine formalism—provides a robust basis for modeling and simulation of discrete event systems. This work was then later expanded to incorporate hybrid discrete event and continuous systems (Zeigler et al. 2000). Having a solid framework for modeling and simulation promoted the concepts of computer-aided M&S (Ören 1982a), simulation engineering, model-based simulation engineering, and simulation-based problem-solving environments (Ören 1996; Yilmaz et al. 2006). Table 1.5 summarizes the relationship between modeling, simulation and their theoretical basis.

Table 1.5 Formal simulation

Issue	Implication to simulation
Separation of concerns (especially model) in simulation	Model-based simulation
Role of system theories	System theory-based simulation
Importance of modeling	Modeling and simulation (M&S)
Model-based activities other than behavior generation	Computer-aided M&S: tools, toolkits, and environments for: modeling, model-base management, symbolic model processing, post-run and post-study processing (analysis, display), etc.
	Simulation engineering Model-based simulation engineering Simulation-based problem-solving environments

Table 1.6 AI-directed simulation

Source of contribution	Implication
Contribution of simulation to AI	Simulation of intelligent entities
Contribution of AI to simulation	AI-supported simulation (for user/system interfaces) AI-initiated simulation AI-monitored simulation

1.4.4 AI-directed Simulation

Synergy of simulation and machine intelligence (or computational intelligence, or Artificial Intelligence [AI]) has been very fruitful. At the beginning, AI benefited from the application of simulation; see for example "Simulation of human thinking" by Simon et al. 1962). Simon posits, "The real power of the simulation technique is that it provides not only a means for stating a theory but also a very sharp criterion for testing whether the statement is adequate." (Simon et al. 1962, p. 123). Hence, simulation-based aspect of Artificial Intelligence was very strong even at its beginning. Possible contribution of AI to simulation was promoted by Ören starting in early 1980s (Ören 1982b, 1995). An outline of AI-directed simulation is given in Table 1.6.

1.4.5 Agent-Directed Simulation (ADS)

Ability of autonomy (or quasi-autonomy) in addition to intelligence results to a very powerful paradigm of agent-directed simulation (see Table 1.7) (Ören et al. 2000; Yilmaz and Ören 2009).

Table 1.7 Agent-directed simulation

Source of contribution	Implication
Contribution of simulation to agents	For simulation of agent systems – Agent simulation
Contribution of agents to simulation	For user-system interfaces: – Agent-supported simulation For run-time activities: – Agent-initiated simulation – Agent-monitored simulation

1.4.6 Soft Simulation

Soft computing is involved in approximate reasoning and function approximation, learning, search, and optimization which are combined in a complementary and synergetic manner (Abraham and Grosan 2005). Soft computing methods enabled to incorporate life vagueness and uncertainty to simulations. Table 1.8 categorizes "soft simulation" where soft computing methods are applied to simulation; and two types of soft simulation are identified, i.e., neural network simulation and fuzzy simulation.

1.4.7 Simulation Systems Engineering

Simulation Systems Engineering (SSE) is the contemporary practice of modeling and simulation studies that emerged through the synergy between systems engineering and simulation during the evolution of simulation. The contribution of simulation to systems engineering has lead us to Simulation-based Systems Engineering (Mittal and Martin 2013; Zeigler et al. 2013; Gianni et al. 2014; Durak and Ören 2016). Simulation systems engineering emerged as systems engineering contribution to simulation studies for large and complex systems. As seen in Table 1.9, simulation systems engineering can be categorized regarding its relation with agent technologies and the characteristics of the simulation system. The utilization of distributed computing in simulation leads us to Distributed Simulation Systems Engineering (DSSE).

Moreover, the synergy between agent technologies, simulation, and systems engineering (Yilmaz and Ören 2009; Ören and Yilmaz 2012) has brought out Agent-directed Simulation Systems Engineering (AdSSE). Its distributed simulation variant is then Agent-directed Distributed Simulation Systems Engineering (AdDSSE).

Table 1.8 Soft simulation

Type of soft computing	Implication to simulation
Neural networks	Neural network simulation
Fuzzy logic	Fuzzy simulation

Table 1.9 Simulation systems engineering (for complex systems and systems of systems)

Without agents	With agents
Simulation Systems Engineering (SSE)*	Agent-directed Simulation Systems Engineering (AdSSE)
Distributed Simulation Systems Engineering (DSSE)	Agent-directed Distributed Simulation Systems Engineering (AdDSSE)

*Distinct from simulation-based systems engineering (for all types of systems engineering)

1.4.8 Simulation-Based Disciplines

Synergies play an important role in the evolution of disciplines. Contribution of simulation to a discipline "x" is called "simulation-based x". Already, simulation-based science, simulation-based engineering as well as many other simulation-based disciplines are important examples of contributions of simulation to other disciplines, making simulation a powerful infrastructure for them. Table 1.10 highlights an extended list of disciplines which benefit tremendously from simulation-based approaches.

Table 1.10 Simulation-based disciplines (examples)

Areas	Disciplines
Engineering	Simulation-based (all types of) engineering (Chaps. 3, 4, 7, 8) Simulation-based cyber-physical systems (Chap. 5) Simulation-based complex adaptive systems (Chap. 6)
Natural science	Simulation-based (all types of) Science (Chap. 9) Simulation-based cosmology Simulation-based astronomy
Health science	Simulation-based Health Care (Chap. 10) Simulation-based pharmacology
Social science and management	Simulation-based Social Science (Chap. 11) (Behavioral science, psychology, demography, sociology, public administration, political science, archeology, environmental studies, …) Simulation-based economics Simulation-based enterprise management (Chap. 12) Simulation-based planning and scheduling Simulation-based optimization Simulation-based policy improvement
Information Science	Informatics Artificial intelligence (machine intelligence) Software agents Communication Library science
Education/training	Simulation-based Education (Chap. 13) Simulation-based training (Chaps. 10, 14) (including health care and military training)
Entertainment	Simulation-based games

In this book, we provide a comprehensive review of these disciplines. The experts of these disciplines present the use of simulation in their discipline and the role of it in the evolution of their discipline. While Chap. 3 introduces simulation in classical engineering process, Chap. 4 presents the role of simulation in systems engineering. In Chap. 7, readers can find a discussion about the role of simulation is software engineering and Chap. 8 explores simulation in architecture. Cyber-Physical Systems (CPS), Internet of Things (IoT), and further complex adaptive systems as emerging systems architectures that come along with a new set of challenges. Simulation is being pronounced as a key tool in tackling the upcoming challenges of engineering. Accordingly, Chap. 5 presents simulation based CPS and IoT and Chap. 6 explores simulation-based complex adaptive systems.

Natural and health sciences are also benefiting from the contribution of simulation in great extent. Chapter 9 presents simulation-based science and Chap. 10 introduces simulation health care, and health education and training. In the field of social sciences and management, there are various applications of simulation. This book includes three chapters that incorporate the social elements in today's sociotechnical systems: Chap. 11 for simulation-based social science and Chap. 12 for simulation-based enterprise management. In the direction of learning, education and training, while Chap. 13 presents simulation-based learning; Chap. 14 presents simulation in military training.

In addition to the disciplines listed in Table 1.10, some other disciplines are already using simulation-based approaches. They include: experimental archeology or simulated dig (Brown) and simulation-based cosmology.

1.4.9 Simulation-as-a-Service

With the computing infrastructures moving to cloud environments, simulation is increasingly being made available as a service. Technically speaking, it is a service for doing "computation"—done to perform experiment(s) or to gain experience—that is invoked through a remote request. However, the computation aspect needs to be supported by a rich and reliable model as well as scenario bases. This enables the end-user to leverage high-performance simulation farms at its disposal to conduct simulation experiments without investing in setting up computing hardware. The end-user typically subscribes for the computational resources needed to perform the computation. The subscription can be individual based or group based. Equally

Table 1.11 Simulation-as-a service

Type of service	Simulation results to be distributed
Individual subscription	On demand Delivered every time there is an update
Group subscription	Individually to members of the group Broadcasting

important is the technical infrastructure that is required to run a simulation in a cloud environment, which is an independent area of research in the current era (Table 1.11).

1.5 Future of Simulation-Based Disciplines

Many of the disciplines outlined in Sect. 1.4 have already accepted model-based approaches and methodologies to support the design of new systems, and analyze the structure and behavior of current systems. However, in some cases, model-based approaches have not yet find their way to simulation-based approaches due to either the lack of need for simulation or project time-constraints, for the simple reason that turning a model-based methodology to a simulation-based methodology is a nontrivial effort—albeit very profitable for the success of the project. The advances in model transformation technologies in the last decade are a step in that direction where abstract models can be turned into concrete models that can be transformed into executable code, and eventually participate in a simulation experiment in the final stage.

This book delineates the benefits of computational environments and emphasizes that modeling and simulation can contribute to future advancements in many disciplines. Not only M&S is benefitting disciplines like sociology, which has largely been insulated from such experimentation, it is undoubtedly used in every aspect of life, whether transportation, finance, economics, biology, and so forth.

Time is ripe for the transition of many disciplines from "model-based" paradigm to "simulation-based paradigm" to make them even more powerful.

Review Questions

1. Give seven reasons why simulation is used. Are they substantial contributions in your field? If not, why?
2. How do you keep up with the advancements of simulation?
   - 2.1 Are you a member of at least one simulation society (as listed at http://www.site.uottawa.ca/~oren/links-MS-AG.htm) to follow developments of simulation in your field?
   - 2.2 How many simulation conferences do you follow every year?
3. Cite about five possibilities how simulation-based approach may be beneficial for your field?
4. What makes a discipline simulation-based? Is simulation used in your field? If not, why?
5. What are the reliability issues of simulation? How can they be assured?
6. Why are ethics important in simulation studies?

7. What is the difference between agent-based simulation and agent-directed simulation? Cite some additional advantages of agent-directed simulation over agent-based simulation.
8. What is the difference between simulation systems engineering and simulation-based systems engineering? Cite advantages of both.
9. Give some examples of soft simulation? In which cases, they can be beneficially applicable?
10. Why exascale computers are important in simulation? Cite some example areas.

Appendix 1.1—Terms Related with Similitude

Simulation is based on the very rich concept of "similitude" which covers a large variety of meanings. In this appendix, terms related with similitude are listed under the following 14 groups: (1) Simulation concept, (2) Model, (3) Analogy, (4) Imitation, (5) Behavioral similarity, (6) Functional similarity, (7) Similarity in mathematics, (8) Similarity in linguistics, (9) Similarity in literature, (10) Similarity in art, (11) To be similar, (12) Indistinguishableness, (13) Disguise similitude under a false appearance, and (14) Non-similarity.

1.1 *Simulation concept*
Auto simulatable
Auto simulate (v)
Auto simulated
Auto simulation
Auto simulative
Co-simuland
Co-simulatable
Co-simulate (v)
Co-simulated
Co-simulation
Co-simulationist
Co-simulative
Meta-simuland
Meta-simulatable
Meta-simulate (v)
Meta-simulated
Meta-simulation
Meta-simulationist
Meta-simulative
Multisimulatable
Multisimulate (v)
Multisimulated
Multisimulation

(continued)

(continued)

Multisimulation-based
Multisimulationist
Multisimulative
Non-simulatable
Non-simulation
Simuland
Simulatable
Simulate (v)
Simulated
Simulating
Simulation
Simulation-based
Simulation-driven
Simulationist
Simulative
Simulator
Simulism
1.2 *Model*
Model
Model (v)
Model-based
Model-driven
Modeler
Modelled
Modeling
1.3 *Analogy*
Alike
Analog
Analogical
Analogous
Analogy
Like
Likeness
Pose (v)
Resemblance
Resemble (v)
Resembled
Resembling
Self-similar
Similar
Similarity
Similitude
Simulacra

(continued)

(continued)

Simulacrum
1.4 *Imitation*
Copy
Imitate
Imitate (v)
Imitated
Imitation
Imitative
Imitator
1.5 *Behavioral similarity*
Mimesis
Mimetic
Mimicry
Pantomime
Pretend (v)
Pretention
Role playing
1.6 *Functional similarity*
Emulate (v)
Emulated
Emulating
Emulation
Emulative
Emulator
1.7 *Similarity in mathematics*
Automorph
Automorphic
Automorphism
Bisimulatable
Bisimulate (v)
Bisimulated
Bisimulation
Bisimulative
Bisimulator
Congruous
Conjugate
Endomorph
Endomorphic
Endomorphism
Endomorphous
Equivalence
Equivalent
Homolog

(continued)

(continued)

Homologic
Homology
Homomorph
Homomorphic
Homomorphism
Homomorphous
Homomorphy
Homothecy
Homothetic
Homothetic transformation
Homothetism
Homothety
Isomorph
Isomorphic
Isomorphism
Isomorphous
Map (v)
Noncongruent
Noncongruently
Strong bisimulation
1.8 *Similarity in linguistics*
Alternative
Equivalence
Equivalent
Homograph
Homographic
Homography
Homonym
Homonymous
Homonymy
Homophon
Homophonous
Homophony
Isomorph
Isomorphism
Synonymous
Synonymy
Tautology
1.9 *Similarity in literature*
Metaphor
Metaphoric
Pastiche

(continued)

(continued)

Pataphor
Pataphoric
1.10 *Similarity in art*
Imitate (v)
Imitation
Pastiche
Replica
1.11 *To be similar*
Assimilate (v)
Assimilated
Assimilating
Assimilatingly
Assimilation
Assimilationism
Homochromy
Homotypy
Mimesis
Mimetic
Mimetism
Mimicry
1.12 *Indistinguishableness*
Indistinguishable
Indistinguishableness
Indistinguishably
Indistinguishing
To be mistaken for
1.13 *Disguise similitude under a false appearance*
Differentiation
Dissimular
Dissimularity
Dissimulate
Dissimulate (v)
Dissimulation
Dissimulative
Dissimulator
1.14 *Non-similarity*
Dissimilar
Dissimilarity
Dissimilarly
Dissimilate (v)
Dissimilation

(continued)

(continued)

Dissimilitude
Non-similar
Unalike
Unique
Uniqueness

References

Abraham, A., and C. Grosan (2005) Soft Computing in Simulation and Modeling, *International Journal of Simulation Systems, Science & Technology*, Vol. 6 No. 10–11, pp. 1–3.

Brown, J. R. and Y. Fehige (2014). Thought Experiments, *The Stanford Encyclopedia of Philosophy* (Spring 2016 Edition), Edward N. Zalta (ed.), http://plato.stanford.edu/archives/spr2016/entries/thought-experiment/

Brown, S. Resources and National Standards for Simulated Digs, Lesson Plans. https://www.archaeological.org/pdfs/education/digs/Digs_resources.pdf

Bruno, G. (1995). Model-based Software Engineering. Chapman & Hall Computing.

Dewey, J. (1934). Art as Experience. Perigee trade paperback edition, 2005, Penguin Group USA.

Durak, U. and T. Ören (2016). Towards an Ontology for Simulation Systems Engineering. Proceedings of the SpringSim'16, Umut Durak and Murat Günal, (eds.), April 3–6, 2016, Pasadena, CA, (SCS – The Society for Modeling and Simulation International), San Diego, CA, pp. 167–170.

(ED – simulation). Online Etymology Dictionary – simulation) [Online]. Available: http://www.etymonline.com/index.php?allowed_in_frame=0&search=simulation [Accessed 2017-02-13].

Gianni, D., D'Ambrogio, A., & Tolk, A. (Eds.). (2014). Modeling and Simulation-Based Systems Engineering Handbook. CRC Press.

Gualeni, S. (2014). Virtual Worlds as Philosophical Tools – How to Philosophize with a Digital Hammer, Springer.

Howe, I. (1979). Literature as Experience: An Anthology. Harcourt College Pub.

JeLSIM (Java eLearning SIMulations). http://www.jelsim.org/training/introduction/simulation.html

Karplus, W.J. (1977). The Spectrum of Mathematical Modeling and Systems Simulation, Mathematics and Computers in Simulation, Vol. 14, pp. 3–10.

Mittal, S., Martin, J.L.R., (2013) *Netcentric System of Systems Engineering with DEVS Unified Process*, CRC Press, Boca Raton, FL USA.

Ören, T.I. (1971). GEST: General System Theory Implementor, A Combined Digital Simulation Language. Ph.D. Dissertation, 265 p. University of Arizona, Tucson, AZ.

Ören, T.I. (1982a– Keynote Paper). Computer-Aided Modelling Systems. In: Progress in Modelling and Simulation, F.E. Cellier (ed.). Academic Press, London, England, pp. 189–203.

Ören, T.I. (1982b). Promising Synergies of Simulation Model Management, Software Engineering, Artificial Intelligence, and General System Theories. In: General Systems, Vol. 26, R.K. Ragade (ed.), pp. 203–209. Also published in: Proceedings of the 26th Annual Conference of the Society for General Systems Research, Vol. 1, pp. 197–204.

Ören, T.I. (1983). Quality Assurance of System Design and Model Management for Complex Problems. In: Adequate Modelling of Systems, H. Wedde (ed.). Springer-Verlag, Heidelberg, W. Germany, pp. 205–219.

Ören, T.I. (1984). Model-Based Activities: A Paradigm Shift. In: Simulation and Model-Based Methodologies: An Integrative View, T.I. Ören, B.P. Zeigler, M.S. Elzas (eds.). Springer-Verlag, Heidelberg, Germany, pp. 3–40.

Ören, T.I. (1995-Invited contribution). Artificial Intelligence and Simulation: A Typology. In: Proceedings of the 3rd Conference on Computer Simulation, S. Raczynski (ed.), Mexico City, Nov. 15–17, pp. 1-5.

Ören, T.I. (1996). Towards Simulative and Intelligent Problem Solving Environments (Position Statement at a Panel on: Synergy of Artificial Intelligence and Simulation). In: Proceedings of 1996 Summer Computer Simulation Conf., Portland, Oregon, July 21–25, SCS, San Diego, CA., p. 596.

Ören, T. (2005). Maturing Phase of the Modeling and Simulation Discipline. In Proceedings of the 2005 ASC- Asian Simulation Conference, Beijing, China.

Ören, T.I. (2010). Simulation and Reality: The Big Picture. (Invited paper for the inaugural issue) International Journal of Modeling, Simulation, and Scientific Computing (IJMSSC) (of the Chinese Association for System Simulation - CASS) by the World Scientific Publishing Co. China, Vol. 1, No. 1, 1–25.

Ören, T.I. (2011a). The Many Facets of Simulation through a Collection of about 100 Definitions. SCS M&S Magazine, 2:2 (April), pp. 82–92.

Ören, T.I. (2011b). A Critical Review of Definitions and About 400 Types of Modeling and Simulation. SCS M&S Magazine, 2:3 (July), pp. 142–151.

Ören, T.I., S.K. Numrich, A.M. Uhrmacher, L.F. Wilson, and E. Gelenbe (2000 - Invited Paper). Agent-directed Simulation: Challenges to Meet Defense and Civilian Requirements. Proc. of the 2000 Winter Simulation Conference, J.A Joines et al., eds., Dec. 10–13, 2000, Orlando, Florida, pp. 1757–1762.

Ören, T.I. and B.P. Zeigler (1979). Concepts for Advanced Simulation Methodologies. Simulation, 32:3, 69–82.

Ören, T.I. and B.P. Zeigler (2012). System Theoretic Foundations of Modeling and Simulation: A Historic Perspective and the Legacy of A. Wayne Wymore. Special Issue of Simulation –The Transactions of SCS. (abstract). Vol. 88, issue 9, Sept. 2012, pp. 1033–1046.

Ören, T., and Yilmaz, L. (2012). Synergies of simulation, agents, and systems engineering. Expert Systems with Applications, vol. 39, no. 1 (January 2012), pp. 81–88. (Available online 6 July 2011).

Ören, T.I., B.P. Zeigler, M.S. Elzas (eds.) (1984). Simulation and Model-Based Methodologies: An Integrative View. NATO ASI Series, Springer-Verlag, Berlin, Heidelberg, New York, 651 p.

Simon, H.A. et al. (1962). Simulation of Human Thinking. In: Computers and the World of the Future, M. Greenberger (ed). MIT press, (2nd paperback printing, 1965), pp. 94–131.

Swaine, M.R., D. Hemmendinger, P.A. Freiberger, and W.M. Pottenger (2917). Computer, Encyclopedia Britannica. Available: https://www.britannica.com/technology/computer [Accessed 2017-04-05].

Tocher, K.D. (1963). The Art of Simulation. English Universities Press, London, UK.

Wallace, A.B. and R.S. Breen (1959). Literature as Experience. McGraw-Hill.

Wymore, A.W. (1993). Model-based Systems Engineering. CRC Press, USA.

Yilmaz, L. and T.I. Ören (eds.) (2009-All Chapters by Invited Contributors). Agent-Directed Simulation and Systems Engineering. Wiley Series in Systems Engineering and Management, Wiley-Berlin, Germany. 520 p.

Yilmaz, L., T.I. Ören, and N. Ghasem-Aghaee (2006). Simulation-Based Problem Solving Environments for Conflict Studies. Simulation, and Gaming Journal. vol. 37, issue 4, pp. 534–556.

Zeigler, B.P. (1976). Theory of Modeling and Simulation (first ed.). Wiley Interscience, New York.

Zeigler, B.P., Praehoefer, H., Kim, T.G., (2000) Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, Academic Press, PA.

Zeigler, B.P., Sarjoughian, H.S., Duboz, R., Soulie, J., (2013) Guide to Modeling and Simulation of System of Systems (Simulation Foundations, Methods and Applications), Springer, UK.

Author Biographies

Tuncer Ören is Emeritus Professor of computer science at the University of Ottawa, Canada. His Ph.D. is in systems engineering from the University of Arizona. He has been active in simulation since 1965. His **research interests** include: (1) advancing modeling and simulation methodologies; (2) agent-directed simulation (full synergy of simulation and agents); (3) agents for cognitive and emotive simulations; (4) reliability, quality assurance, failure avoidance, and ethics; as well as (5) body of knowledge and (6) multilingual terminology of modeling and simulation. Dr. Ören has over 500 **publications**, including 40 books and proceedings. He has contributed to over 500 **conferences** and seminars (about half as invited/honorary) in about 40 countries. He was inducted to SCS Modeling and Simulation Hall of Fame–Lifetime Achievement **Award**. He is also a distinguished lecturer, an SCS fellow, and AVP for ethics. He is recognized, by IBM Canada, as a pioneer of computing in Canada. A book about him was edited in 2015 by Prof. L. Yilmaz: Concepts and Methodologies for Modeling and Simulation: A Tribute to Tuncer Ören, Springer. **Other distinctions** include invitations from United Nations and plaques and certificates of appreciation from organizations including ACM, Atomic Energy of Canada, and NATO.

Saurabh Mittal is Lead Systems Engineer/Scientist at MITRE, founder and President of Dunip Technologies, LLC and Vice President-Memberships for Society of Computer Simulation (SCS) International. He is also currently serving as Associate Editor-in-Chief for Transactions of SCS and Editor-in-Chief of Enterprise Architecture Body of Knowledge (EABOK) Consortium. He received his M.S. and Ph.D. in Electrical and Computer Engineering from the University of Arizona. Previously, he was a scientist and architect at National Renewable Energy Laboratory, Department of Energy at Golden, Colorado, where he contributed to complex energy systems co-simulation environments. He also worked at L3 Link Simulation & Training as a contractor to US Air Force Research Lab at Wright-Patterson Air Force Base, Ohio where he integrated artificial agents and various cognitive architectures in Live, Virtual and Constructive (LVC) environments using formal Systems theoretical model-based engineering approaches. He was a Research Assistant Professor at the Department of Electrical and Computer Engineering at the University of Arizona. Dr. Mittal served as General Chair of Springsim'17 and SummerSim'15, Vice General Chair for SpringSim'16 and SummerSim'14, and Program Chair for SpringSim'15. He is a recipient of US Department of Defense (DoD) highest civilian contraction recognition: Golden Eagle award and SCS's Outstanding Service award. He has co-authored nearly 70 articles in various international conferences and journals, including a book titled "Netcentric System of Systems Engineering with DEVS Unified Process", CRC Press in 2013 that serves the areas of executable architectures, service-oriented distributed simulation, System of Systems engineering using Department of Defense Architecture Framework (DoDAF), multiplatform modeling, intelligence-based, complex, adaptive and autonomous systems, and large scale M&S integration and interoperability.

Umut Durak is a research scientist in the Institute of Flight Systems at the German Aerospace Center. He is also Adjunct Lecturer in the Department of Informatics at Clausthal University of Technology. His research interests include model-based simulation engineering, simulation-based systems engineering and ontologies in simulation. He received his BS, MS and Ph.D. degrees in Mechanical Engineering from Middle East Technical University (METU) in Turkey. He is a member of the Society for Computer Simulation International (SCS) and Arbeitsgemeinschaft Simulation (ASIM) and a senior member of the American Institute of Aeronautics and Astronautics (AIAA). And, he is a part of AIAA Modeling and Simulation Technical Committee and editorial teams of the Simulation: Transactions of the Society for Modeling and Simulation International; and the International Journal of Modeling, Simulation, and Scientific Computing.

Chapter 2
Modeling and Simulation (M&S) Technology Landscape

Ernest H. Page

Abstract A review of current investment levels in M&S research, development and application is provided, and a subjective assessment of the "leading" organizations across various applications of M&S is suggested. In addition, a number of challenge problems in M&S are identified. Our objective is to provide a starting point for organizations in their formulation of investment and technology strategies for M&S.

Keywords Analysis · Aviation · Defense · Experimentation · Healthcare · Immersion · Live-virtual-constructive · Manufacturing · Research and development · Systems design · Technology landscape · Training

2.1 Introduction

As the Table of Contents for this book suggests, Modeling and Simulation (M&S) is essentially ubiquitous across the scientific and engineering disciplines. As such, holistic, comprehensive treatments of the subject are elusive. In this chapter, we consider the technology investment "landscape" for M&S. Based on publicly available information, we characterize the interest in, and reliance on, M&S—as

The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved for Public Release: Case 17-1282.

E.H. Page (✉)
Modeling, Simulation, Experimentation and Analytics Technical Center,
The MITRE Corporation, Mclean, USA
e-mail: epage@mitre.org

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_2

measured in terms of investment—across various domains of application, across industries, countries, and so forth. Our treatment is necessarily abbreviated and approximate, and superficial in many aspects. Certainly, more extensive treatments of this subject are warranted as the "profession" of modeling and simulation emerges. Nonetheless the cursory examination here may provide a useful starting point for business leaders and/or governmental organizations in their formulation of technology investment and research strategies.

2.2 The Global M&S Landscape

Modeling and simulation pervades science and engineering, with application in systems design and analysis, training, experimentation, mission rehearsal, test and evaluation, education and entertainment. It has been suggested (Glotzer et al. 2009) that:

> Today we are at a 'tipping point' in computer simulation for engineering and science. Computer simulation is more pervasive today – and having more impact – than at any other time in human history. No field of science or engineering exists that has not been advanced by, and in some cases transformed by, computer simulation. Simulation has today reached a level of predictive capability that it now firmly complements the traditional pillars of theory and experimentation/observation. Many critical technologies are on the horizon that cannot be understood, developed, or utilized without simulation.

Despite, or perhaps because of, this ubiquity, global investment levels in M&S are difficult to quantify with accuracy. In most circumstances, we can only measure M&S investments indirectly as a fraction of the total government, industrial, and academic investments across the scientific and engineering disciplines. In some areas, though, distinct M&S marketplaces exist, most notably those involving training simulations/simulators and Product Life-Cycle Management (PLM) software (a cornerstone of the manufacturing industry). Rigorous market analysis data is available in a few of the domains associated with these technologies, including:

- Defense training and simulation (Frost and Sullivan 2014; Visiongain 2015)
- Civil naviatio training and simulation (TechNavio Infiniti Research Ltd 2014)
- Manufacturing (CIMdata 2014)
- Healthcare (Marketsandmarkets 2014; Meticulous Research 2014)
- Emergency Management (Marketsandmarkets 2013).

Although these applications constitute a small fraction of the total M&S landscape, collectively they represent an estimated global annual market exceeding $18B USD. The countries with the highest investment levels in training simulation and simulators include: United States, Russia, China, India, and United Kingdom. The heaviest investors in PLM software are: France, United States, and Germany. Asia-Pacific and Latin America are expected to have the highest growth in medical simulation, driven by: India, China, South Korea, Singapore, Brazil, and Mexico.

An important subset of total spending is investments relating to research and development (R&D). Within the government sector, M&S R&D investments are typically embedded within the enterprise Science and Technology (S&T) budget, or as part of the Research, Development, Testing and Experimentation (RDT&E) budget for new/developing systems. Again, direct measures are elusive, but within the U.S. alone, annual R&D spending easily exceeds $100B USD (Valvida and Clark 2015). An informal scan of the programs across the Department of Defense, Department of Energy, National Science Foundation, National Air and Space Administration, National Institute of Standards and Technology, National Institutes of Health, National Labs, and Federally Funded Research and Development Centers (FFRDCs) suggests that tens of billions (USD) are oriented toward M&S-related topics annually. Global expenditures may eclipse that value by an order of magnitude.

With respect to investments by industry, in-depth market surveys for global industrial R&D funding are available. A 2016 assessment suggests that total global R&D investments approach $2T USD (Industrial Research Institute 2016). Asian countries (including China, Japan, India, and South Korea) account for more than 40% of the total investments; North America represents 30%, Europe 20%, and the rest of the world (Russia, Africa, South America, and the Middle East countries) account for 10%. Again, we can only estimate M&S as a fraction of this total R&D spending. However, the 2016 study cited above specifically identifies M&S as a critical R&D technology.

Given the value and volume of M&S workloads within scientific computing, another indirect measurement for M&S R&D investment may be the Top500 list of supercomputing sites. For 2016, countries represented in the top 25 are: China, U.S., Japan, Switzerland, Germany, Italy, France, and Saudi Arabia (Top500.org 2016).

From the data in these formal market studies, in combination with less formal assessments of the scientific literature, and the activities of scientific and professional societies, we derive a partial (and largely subjective) view of international leadership in M&S, summarized in Table 2.1. Here, "leadership" is simply an aggregated function of investment levels, publication volumes, and subjective measures of influence, notoriety, and so forth. Obviously, many important entries may be missing from this data. The entries in Table 2.1 merely suggest a starting point for any rigorous analysis relating to M&S technology investment and research strategies.

Table 2.1 A partial view of international leadership in M&S

Nation	Leading govt. organizations	Leading academic institutions	Industry leaders
UK	DSTL iDSC	Brunel Univ. University of Edinburgh Imperial College Loughborough University University of Southampton	Rolls-Royce QinetiQ Saker Solutions Simul8 Corp
Germany	Max Planck Inst.	University of Rostock University of Munich University of Stuttgart ASIM	Fraunhofer Inst. Siemens SAP Volkswagen
France	European Space Agency	INSEAD Ecole Normale Ecole Centrale Ecole des Mines St-Etienne Supérieure, Paris	Thales Dassault Systems Renault Airbus
Russia		Lomonosov Moscow State University	AnyLogic
China	Chinese Assoc. for Systems Simulation	China University of Science and Technology	
Canada		McGill University Carleton University University of Ottawa University of Calgary	Lumerical Bombardier Thales CAE Autodesk Research
Singapore	Defence, Science and Technology Agency	Nanyang T.U.	
Netherlands		Tilburg University TU Delft	

2.3 The U.S. M&S Landscape

As with global investment levels, quantifying U.S. investments in M&S is also difficult. The U.S. uses the North American Industry Classification System (NAICS) to classify business activity in the nation. Despite the notable efforts of groups like SimSummit (www.sim-summit.org), an international consortium of M&S entities across government, academia and industry, to facilitate the creation of a NAICS code(s) for M&S, none has yet been defined. Therefore, M&S activity must largely be measured indirectly.

A 2012 study established total U.S. expenditures on M&S at $50B USD annually, including $9B USD within the Department of Defense (DoD) (Old Dominion University 2012). States with significant activity in M&S, including dedicated research centers, include: Virginia, Florida, Arizona, California, and Alabama. While direct measures of M&S R&D activity are unavailable for this analysis, the total number of articles associated with the keywords "modeling" or "simulation" available within the major digital libraries (ACM, IEEE Xplore, etc.) is increasing. The number of venues for research publication (conferences, workshops, journals) also seems to be increasing. And the number of Universities granting graduate degrees in M&S continues to rise.

M&S is a topic of interest at the highest levels of the U.S. government. In June 2007, the U.S. House of Representatives approved House Resolution 487, which identifies M&S as a National Critical Technology. This resolution was developed through the M&S Congressional Caucus under the direction of Congressman J. Randy Forbes (4th District VA), and establishes the importance of M&S to the national security.

In a July 14, 2010 statement to the Subcommittee on Commerce, Trade and Consumer Protection of the Committee on Energy and Commerce, U.S. House of Representatives, Aneesh Chopra, the Chief Technology Officer and Associate Director of the Office of Science and Technology Policy, Executive Office of the President, asserted that M&S can significantly reduce the need for physical prototypes in the manufacturing sector of the U.S. economy. This, he said, would shorten product development time, reduce costs, and improve quality. Chopra believes that M&S is capable of providing the country with a crucial manufacturing edge that will lead its manufacturing renaissance (Old Dominion University 2012).

In fall 2011, the National Modeling and Simulation Coalition (NMSC) was formed (www.modsimcoalition.org). The mission of the NMSC is to create a unified national community of individuals and organizations around the M&S discipline and professional practice and to be the principal advocate for national investments in M&S.

2.4 Some "Good Challenges" in M&S

Over the past decade and a half there has been significant energy in the identification and description of "Grand Challenges" for M&S (Taylor et al. 2013; Fujimoto et al. 2017). The community has done a great service in collectively generating and vetting a wide range of thoughtful and impactful fundamental research challenges. When confronted with a Grand Challenge, you generally know where to begin to look for the funding and intellectual capacity necessary to attack it, e.g., NSF, DARPA, major research institutions, etc. But what about the "lesser" challenges? The semi-formal market assessment described above was undertaken, in part, to support the development of a research strategy with a distinctly "applied" focus. We include some of those challenge areas here—which we'll call "Good Challenges"—along with their alignment to some of our identified market leaders.

We consider the application of M&S in three principal areas:

- Systems design and analysis
- Training, experimentation, and mission rehearsal
- Testing and integration.

Collectively, these areas present an interesting spectrum of technical challenges for M&S including: execution mode (standalone, human-in-the-loop, hardware-in-the-loop, real time, faster-than-real time), implementation language language(s), data management approaches, statistical methods, visualization, abstraction, and fidelity, verificationand validation, and reasoning about uncertainty and risk.

For systems design and analysis, one area of focus is **the application of high performance and ubiquitous computing, multi-model integration, advanced analytics, and visualization to support strategic-level decision-making in complex environments**. Topics of interest include:

- **Simulation-based optimization**. Within the government (and also in surprisingly many industrial settings) systems design analyses often find their basis in small set of "blessed" scenarios, and involve a fairly small number of design points. An opportunity exists to help decision makers embrace optimization-based methods—particularly those where automated support is available. Such methods are essential, for example, to the engineering of agile systems.
- **Metamodeling**. Robust analysis, typically supported by long-running experiments using high-fidelity models, is an essential component of good systems design and analysis. Making the results of such studies understandable to senior strategic-level decision-makers can be a challenge. One approach, may be through the use of metamodels generated from high-fidelity models. Allowing senior decision-makers to interact in realtime with reasonably accurate metamodels (and their visual representations) may facilitate better understanding of a system and its responses.
- **Immersive visualization**. Another approach to the problem of communicating the results of complicated models to senior-level decision-makers is through visualization. Can we develop visualization techniques that "immerse" a decision maker in the model and its results? Does such immersion lead to increased understanding and better insights? What modes of interaction with the model results can we provide? What are their relative effectiveness?
- **Ensemble modeling**. It is sometimes forgotten that a model is simply an *opinion* about the way the world works. If you are making critical decisions, you probably could benefit from having more than just one opinion regarding your course of action. Budgetary pressures within the government generally result in a narrowing of the model marketplace—there is generally an appetite for singular, definitive, models of any given phenomenon. There is an opportunity to help decision makers understand the value of ensemble modeling. An extension of this concept is *generalized crowd-sourcing* (predictive markets) by which multiple opinions/agendas are synthesized.

- **Prospective analytics**. A computer cycle is a terrible thing to waste. How should an organization take advantage of its intrinsic computing capabilities to exercise models and analytics in *anticipation* of questions a customer/sponsor may ask?
- **Merging M&S and big data analytics**. A fundamental tenet of M&S is that a model must be built with a specific purpose in mind (i.e., a specific set of questions that the model is intended to answer). However, the emergence of big data analytics may offer a challenge to this old way of thinking. What if we simply set out to create "models of the world"—representing entities and relationships as we perceive the need—and use these models to generate time-series data relating to every entity represented in the model and then apply big data analytics to the output? Does this add useful flexibility to our analytic processes?
- **Quantifying uncertainty and risk**. Computing and accumulating approximate error is part and parcel of many continuous modeling techniques, but discrete event methods (to include agent-based methods) are largely silent on this. In addition, we need better ways of relating these uncertainty measure to underlying risk.

For training, experimentation and mission rehearsal, one area of focus is **the application of immersive technologies (virtual reality, augmented reality, telepresence, visualization, synthetic environments, virtual humans) LVC integration, and low-overhead, high-automation techniques to produce low-cost/high-value environments**. Topics of interest include:

- **Virtual reality**. The positive impact on immersion on the effectiveness of simulation-based training and experimentation is well known (although "how much immersion is enough?" remains an open question). The contributions of VR technology to immersion are also well-known. As the commercial VR marketplace continues to grow, the interest in applying these commercial technologies in non-gaming contexts increases.
- **Augmented reality**. A longstanding pursuit within the military simulation community is the definition and development of architectures and technologies that enable the integration of Live, Virtual and Constructive (LVC) elements within a single, concurrent event (for training, experimentation or mission rehearsal), the effective use of AR to allow Live participants to perceive events generated by Virtual and Constructive components is needed. Unlike VR technologies, however, the commercial market for AR is waning—assessed by (Gartner 2014) to be in the "trough of disillusionment".
- **Virtual humans**. Role players are a part of most medium- to large-scale training and experimentation events. However, their presence can decrease the immersive nature of the experience, and can also introduce errors. The use of *virtual humans*—computer-generated characters that use language, have appropriate gestures, show emotion, react to verbal and nonverbal stimuli—has the potential to provide a low-cost, highly effective solution to the problems associated with

Table 2.2 A partial view of leadership in M&S (by selected topic area)

Topic	Leading gov. organizations	Leading academic institutions	Industry leaders
Defense	TRADOC NAWC TSD PMTRASYS AFAMS AMSO SIMAF RAND AFRL ARL MITRE Lincoln Labs	UCF/IST NPS AFIT GMU	NTSA NDIA Aegis VT MaK Boeing Raytheon Lockheed Martin Roland and Assoc
Aviation	AFSOR JPL NASA	MIT Caltech Stanford University of Michigan Georgia Tech	Boeing Lockheed Martin L3
Networking (cyber)		Ga Tech University Illinois FIU	Cisco Riverbed SNT
ISR	Aerospace LLNL	AFIT Carnegie Mellon	AGI Terra Bella Black Sky Global
Experimental design	RAND AFRL ARL	Northwestern, Cornell, Georgia Tech, NCSU, NPS, Tilburg Univ	Boeing, Fraunhofer Inst, Phoenix Integration
Optimization	RAND ORNL	Northwestern Cornell Georgia Tech	Boeing Google
High performance and ubiquitous computing	ORNL, LANL, LLNL, ANL, ARL, Sandia, NASA Ames, AFRL	Caltech, MIT, University of Illinois Urbana-Champaign, University Texas, Edinburgh University Georgia Tech, Virginia Tech	IBM, Cray, Google, Amazon
Immersive technology	Training Brain Ops Center	USC/ICT USF/IST	Redfish Occulus

(continued)

Table 2.2 (continued)

Topic	Leading gov. organizations	Leading academic institutions	Industry leaders
LVC integration	PEO STRI, MITRE	UCF/IST, Nanyang Technical University	Raytheon, Aegis, VT MaK, Lockheed Martin, NTSA
Embedded systems and hardware-in-the-loop	JPL, ARL	T.U. Delft, Arizona State, Georgia Tech, Carnegie Mellon University	Siemens, Boeing, Rolls-Royce, Intel

role players (Institute for Creative Technologies 2017). In addition, the use of such characters can significantly extend the range and scope of a given training or experimentation event.

- **Low-overhead event support**. In addition to the overhead associated with role players, most training and experimentation events have considerable overhead in "technical support". The provision for such technical support is a major impediment to the U.S. DoD's (among other major institutions) ability to fully realize its vision for Home Station Training (Perkins 2012).

For integration and testing, one area of focus is **the development and application of high assurance environments for system evaluation that support moving from a paradigm of "test-based confidence" to "simulation-based confidence"**. Topics of interest include:

- **Large-scale emulation**. Many cyber network effects, for example, cannot be studied at small scales. Further, the network representations much be extremely high fidelity to be effective.
- **Statistics of small samples**. A longstanding problem for the Test and Evaluation (T&E) communities. How can mathematics and statistics be most usefully applied in environments where the number of experimental trials is necessarily small? The integration of nonparametric statistics, applied asymptotics, etc., within our M&S toolkits is of interest.
- **High assurance synthetic environments**. Today's synthetic environments and virtual worlds, largely driven by the commercial gaming market, do not typically represent real-world physics in a manner sufficient to provide engineering-level analysis and evaluation.

As with our table above characterizing global leadership in M&S, Table 2.2, below presents a partial view of leadership in M&S across a variety of topic areas, with an obvious bias toward U.S. entities. Once again, "leadership" in this context is simply an aggregated function of investment levels, publication volumes, and subjective measures of influence, notoriety, and so forth. The entries in Table 2.2 merely suggest a starting point for any rigorous analysis relating to M&S technology investment and research strategies; many important entries may be missing from this data.

2.5 Summary

Due to its pervasiveness across the scientific and engineering disciplines, comprehensive treatments of M&S are difficult to construct. Nonetheless, organizations charged with defining research and technology investment strategies should always do so with a general sense of the research and technology investment strategies of both their competitors and partners. In this chapter, we present a necessarily approximate view of the M&S technology investment landscape. Our survey methodology is, at best, quasi-scientific. We cite formal market surveys and analysis where they exist, but note that these surveys only cover certain segments of the M&S domain space. Other aspects of our treatment are based on informal assessments of the scientific literature and the activities of professional societies, industrial consortia, and so forth. Many of the conclusions here are subjective. Despite these notable weaknesses, the information provided may prove a useful starting point for organizations conducting research and technology investment planning.

Review Questions

1. What are the estimated global investment levels in M&S and M&S-related technologies? How accurate can such estimates be?
2. Which regions spend the most on R&D?
3. What are the major sources of R&D funding by country/region?
4. What topics in M&S might we expect to see increasing investment in over the near-to-mid-term?

References

CIMdata. (2014). *2014 Simulation and Analysis Market report.* Ann Arbor, MI: CIMdata.

Frost and Sullivan. (2014). *Global Military Training and Simulation Market Assessment.* New York, NY: Frost and Sullivan.

Fujimoto, R., Bock, C., Chen, W., Page, E., & Panchal, J. (2017). *Research Challenges in Modeling and Simulation for Engineering Complex Systems.* Springer-Verlag: London, UK.

Gartner. (2014, August). Retrieved from Gartner's 2014 Hype Cycle for Emerging Technologies Maps the Journey to Digital Business: http://www.gartner.com/newsroom/id/2819918?fnl=search.

Glotzer, S. C., Kim, S., Cummings, P., Deshmukh, A., Head-Gordon, M., Karniadakis, G., . . . Shinozuka, M. (2009). *International Assessment of Research and Development in Simulation-Based Engineering and Science.* Baltimore, MD: World Technology Evaluation Center, Inc.

Industrial Research Institute. (2016). 2016 Global R&D Funding Forecast. *R&D Magazine*.
Institute for Creative Technologies. (2017, March). Retrieved from Virtual Humans project: http://ict.usc.edu/groups/virtual-humans/.
Marketsandmarkets. (2013). *Incident and Emergency Management Market – Worldwide Market Forecasts and Analysis 2013-1018*. London, UK: MarketsandMarkets.
Marketsandmarkets. (2014). *Healthcare/Medical Simulation Market by Product & Services (Patient Simulator, Task Trainers, Surgical Simulator, Web-based Simulation, Software, Dental Simulator, Eye Simulator), End User (Academics, Hospital, Military) - Global Forecast to 2019*. London, UK: Marketsandmarkets.
Meticulous Research. (2014). *Global Virtual Patient Simulation Market Outlook: 2014-2019*. Middlesex, UK: Meticulous Research.
Old Dominion University. (2012). *Modeling and Simulation in Hampton Roads*. Norfolk, VA: Available at: https://www.odu.edu/content/dam/odu/offices/economic-forecasting-project/docs/2012-sor-modelsim.pdf.
Perkins, D. (2012, February). Retrieved from Futiyre of Home Station Training: https://www.youtube.com/watch?v=3Yn4KXg2fdc.
Taylor, S., Brailsford, S., Chick, S., L'Ecuyer, P., Macal, C., & Nelson, B. (2013). Modeling and Simulation Grand Challenges: An OR/MS Perspective. *Winter Simulation Conference*. Washington, D.C.: IEEE.
TechNavio Infiniti Research Ltd. (2014). *Global Civil Aviation Flight Training and Simulation Market: 2014-2018*. London, UK: TechNavio.
Top500.org. (2016, Nvember). *Top 500 List*. Retrieved from https://www.top500.org/lists/.
Valvida, W. D., & Clark, B. Y. (2015). *The Politics of Federal R&D: A Punctuated Equilibrium Analysis*. Washington, DC: Brookings Institute.
Visiongain. (2015). *Military Simulation, Modelling and Virtual Training Market Report 2016-2026*. London, UK: Visiongain.

Author Biography

Ernest H. Page is the Chief Engineer for the Modeling, Simulation, Experimentation and Analytics Technical Center at MITRE Corporation, McLean, VA. He received the Ph.D. in Computer Science from Virginia Tech in 2004. The author of over 50 articles in the areas of simulation methodologies and distributed computing, he currently serves on the editorial boards for SCS Simulation, the Journal of Defense Modeling and Simulation, and Journal of Simulation. Previously, he served as Chair of the ACM Special Interest Group on Simulation (1999–2001), and as a member of the Winter Simulation Conference Board of Directors (2001–2012).

Part II
Engineering and Architecture

Chapter 3
Simulation-Based Engineering

Melih Cakmakci, Gullu Kiziltas Sendur and Umut Durak

Abstract Engineers, mathematicians, and scientists were always interested in numerical solutions of real-world problems. The ultimate objective within nearly all engineering projects is to reach a functional design without violating any of the performance, cost, time, and safety constraints while optimizing the design with respect to one of these metrics. A good mathematical model is at the heart of each powerful engineering simulation being a key component in the design process. In this chapter, we review role of simulation in the engineering process, the historical developments of different approaches, in particular simulation of machinery and continuum problems which refers basically to the numerical solution of a set of differential equations with different initial/boundary conditions. Then, an overview of well-known methods to conduct continuum based simulations within solid mechanics, fluid mechanics and electromagnetic is given. These methods include FEM, FDM, FVM, BEM, and meshless methods. Also, a summary of multi-scale and multi-physics-based approaches are given with various examples. With constantly increasing demands of the modern age challenging the engineering development process, the future of simulations in the field hold great promise possibly with the inclusion of topics from other emerging fields. As technology matures and the quest for multi-functional systems with much higher performance increases, the complexity of problems that demand numerical methods also increases. As a result, large-scale effective computing continues to evolve allowing for efficient and practical performance evaluation and novel designs, hence the enhancement of our thorough understanding of the physics within highly complex systems.

M. Cakmakci (✉)
Department of Mechanical Engineering, Bilkent University, Ankara, Turkey
e-mail: melihc@bilkent.edu.tr

G. Kiziltas Sendur
Mechatronics Engineering Program, Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
e-mail: gkiziltas@sabanciuniv.edu

U. Durak
Institute of Flight Systems, German Aerospace Center (DLR), Braunschweig, Germany
e-mail: umut.durak@dlr.de

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_3

Keywords Engineering design cycle · V-process · Waterfall model · Hardware-in-the-loop simulations · Feature-in-the-loop simulations · Component-in-the-loop simulations · Continuum mechanics · Computational electromagnetics · Partial differential equations (PDE) · Finite element method (FEM) · Finite-difference method (FDM) · Multi-scale methods · Lumped parameter models · Model-based control system design · Vehicle dynamics models · Networked control systems · Discretized systems · Quantization · Observer models · Iterative learning

3.1 Introduction

3.1.1 Overview of the Engineering Design Process

The ultimate objective of all engineering projects is to reach a functional design without violating any of the performance, cost, time, and safety constraints often optimizing the design for one of them. Generally, in the beginning of each project high-level requirements for the system is developed. These high-level requirements can be as literal as "*The fuel consumption of the vehicle shall be* 40 mpg *or more*." or comparative such as "*The new CNC machine will be as precise as our competitors*." Then, these high-level requirements are cascaded down to the lower levels of the system design steps to obtain the well-defined engineering design problems.

Engineering design problems are concrete problem constructs that contain quantifiable performance and constraint metrics. The inputs to the engineering problems are the performance constraints, design parameters, external conditions. The output of the engineering design process is the design communicated in technical terms such as materials, dimensions, and algorithms. Usually, a lesser focused output of an engineering project is the operation recommendations, lifecycle maintenance, and storage instructions. In general, main steps of the engineering design process can be given as, requirements analysis, design, implementation, verification, and maintenance.

In Fig. 3.1, inputs and outputs of the engineering development process is given as discussed previously. It is also important to note that this process can be applied at the component, the sub-system (i.e., group of interrelated components), and at the system level.

Over time, two primary approaches emerged to approach the solution of complex engineering design projects.

The early approach also known as the "Waterfall Design Process", the sub-problems can be tackled and solved sequentially. Even though, it provides a structured method to perform design and testing tasks, its sequential nature fails to catch design-related errors early in the development process.

Inspired from the approaches in development of software intensive systems, as an extension of "Waterfall Design Process", a new engineering design approach has emerged which is called the "V-process" where relations of design and validations

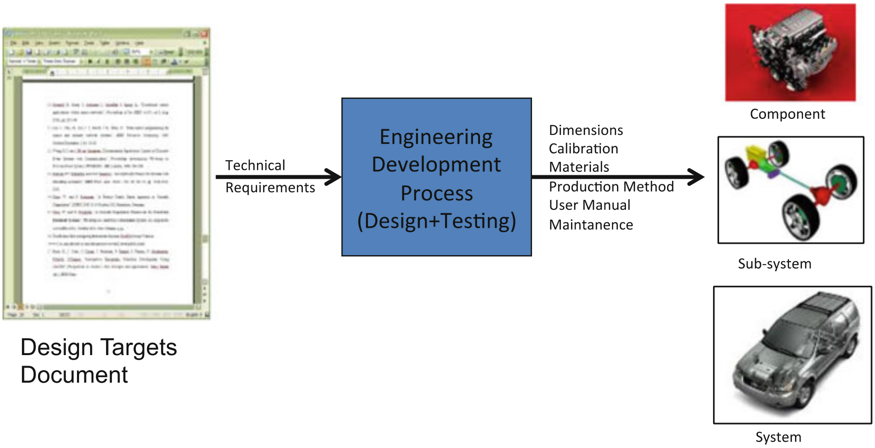


Fig. 3.1 Engineering development process

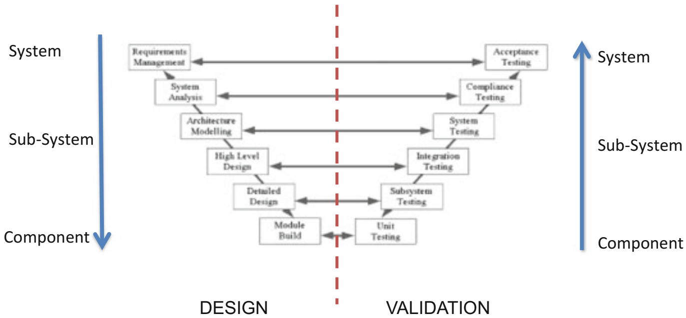


Fig. 3.2 Engineering V-process

steps are stressed. Simulations of varying resolution and fidelity become important tool in the V-process, in order to conduct validations as well as the evaluations of design decisions before the actual prototype of the system can be build.

In Fig. 3.2, the typical steps of the V-process is given based on (Ulsoy et al. 2012). The essence of the V-process is to cascade from the system level to the smaller scale such as the component level and the level-based validation of the work to catch problems at early stages. The different levels of validation and design work in the V-process increase the importance of effective simulations throughout the whole process.

Today almost all of the engineering community is using the iteration based V-diagram process. One of the early hesitation points regarding the engineering V-process was also its strongest feature, namely the existence of stepwise iterations and the cost they bring to the overall development. However, the evolution of advanced simulation techniques made iterations more manageable minimizing the cost of rework during development.

One of the most important topics in the simulation development process is the decision of the feature content and their fidelity. Too much content or dynamics and the simulation will be consuming too much computational resources generally resulting in time and cost problems. Too little detail in simulations will result in misguided simulations, missing important modes of the target systems and taking away the benefit of simulation-based iteration in the development process.

When the engineering V-process is considered, the level of validation increases as the project progresses in time as shown in Fig. 3.2. The decomposition of the requirements and development process requires a proof-of-concept simulation first, which are detailed in physics but abstract at the interface level. When the test phase starts these component-based simulations are combined to produce sub-system and system-level simulations that are also advanced in terms of the mechanical and electronics interactions (the interface) of the system. The sub-system and system-level simulations are more detailed than the component-level simulations.

In most engineering development projects, the understanding of the target system improves with the progress of the project. Therefore, the resolution and fidelity of the simulations can also be improved using the new data and understanding of system of interest.

Generally, in the early stages of the engineering development process a prototype of the target system does not exist. However, a concept emulating simulation of the system can be developed using existing models from the company's resources or from the existing technical literature. When these simulations are functional, the new feature of the system can be included in the model and the simulations can be used to make early predictions about the performance of the target system with fairly good confidence. These simulations can be used to verify the feature-based requirements in the V-process. These simulations are usually known as feature-in-the-loop simulations.

Figure 3.3 shows a simulation case where a new feature (Feature A) in the system is simulated with the already validated features (Features B-E). Even though the system (Features A-E) may have more than one component, in feature-in-the loop simulations the physical boundaries and interfaces are not considered.

After enough confidence is gained about the feature-in-the-loop simulations for the new features of the components, component-in-the-loop simulations can be developed. Component-in-the-loop simulations usually contain all the newly developed features of the system. Feature-in-the-loop simulations are generally done separately for easy troubleshooting and for de-coupling of individual contributions.

Component-in-the-loop simulations contain all the new and carry-over features as well as the actual electronic and mechanical interface of the system. These simulations are used in the component-level testing for the engineering development process. When prepared properly with the actual system-level interface they can be directly used in system simulations that include all components of the system both electric and mechanical.

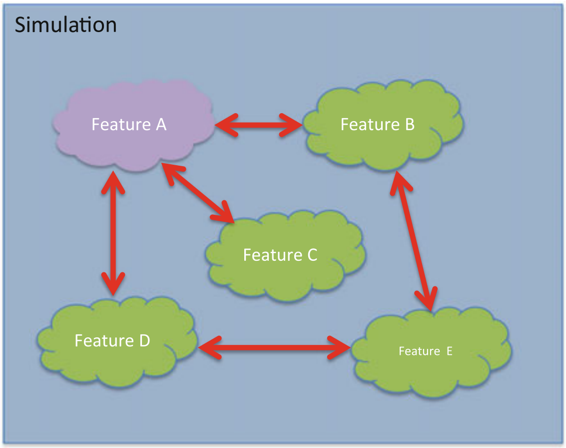


Fig. 3.3 Feature-in-the-loop simulations

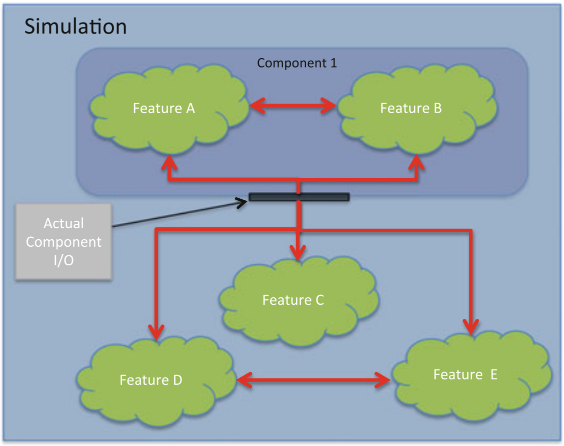


Fig. 3.4 Component-in-the-loop simulations

In Fig. 3.4, a component-in-the-loop simulation scenario is given. The work is done all in the simulation environment, however, a notion of the component physical boundaries exists that forces the interaction of Features A and B through a common interface as compared to the feature-in-the-loop simulation given in Fig. 3.3. Generally, this interface is developed as the proposed physical and electrical interface of the component with the rest of the system.

One of the most important challenges of an engineering development process is to work in a task-based team environment, where different teams are in charge of different features/components/sub-systems of the project. The development cycle of different targets can be at different stages at different times, which makes it difficult to validate functionality with the complete configuration of the system. Testing with hardware-in-the-loop simulations is an approach developed by engineers to overcome this problem. In hardware-in-the loop simulations part of the system is emulated using computers using simulations and part of the system is the actual hardware, which already designed or carried over from the previous version of the system. In many cases the benefit of the HIL simulations are bidirectional in the sense that they can both be used for improving the quality of the simulations using it against the actual hardware or testing a specific prototype hardware for functionality while emulating the rest of the system.

Figure 3.5 shows a hardware-in-the-loop scenario for the system and features given in Figs. 3.3 and 3.4. This time the actual hardware of the component that includes Features A and B are run against the rest of the system (Features C–E) all simulated in the computer environment. It is also important to note that the preparation of the simulations in the earlier stages help to build successive versions of the feature, component and hardware-in-the-loop simulations. For example the physical and electrical based build of the interface in the component level increase the reuse of the component representation in the hardware-in-the-loop simulations.

A good example of the simulation-based V-process development is the so-called mode-based controller development process (MBCD) in the automotive industry. In (Ulsoy et al. 2012), a technical requirements development method is shown for a specific battery control module example. This example shows how the vehicle 100,000-mile requirement affects specific features (control problems) for a particular vehicle application. The effect of this requirement and others define the feature control problem to be solved. The solutions obtained from all of the features represent the control algorithm for a vehicle.

In the design step, first the control design problem is formulated based on the given performance requirements and developed mathematical formulation. There will be more than one control design approach, which will provide a solution for the control problem. By using analytical methods and/or computer simulations, the best alternative among these candidate algorithms is selected. If the control problem is similar to an earlier application, development teams often prefer to start with an

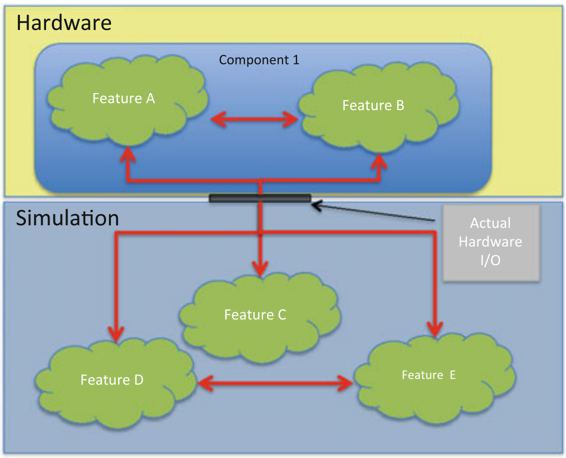


Fig. 3.5 Hardware-in-the-loop simulations

existing control algorithm and try improve the solution by building upon the existing (and proven) solution.

Then the design is implemented on the actual hardware. During the implementation phase the objective is to develop a real-time application, which will be executed in the control module using the desired control algorithm. While developing the executable code the real-time constraints of the target hardware (i.e., the controller module) should also be considered. Software implementation of the algorithm should be matched to the computing resources available and if there are overruns during the real-time execution simplifications in the algorithm should be made, or new target hardware should be selected. In today's modern vehicles, controller modules also communicate with other controllers via communication networks. The effects of the loss of this communication with one or more contacts or the cases of limited communications should be investigated and necessary modifications should be made.

Testing in the MBCD process starts as early as in the algorithm development step. By testing the algorithms open-loop (Fig. 3.6a) developers can feed in simple test vectors and analyze the test output for expected functionality. These simple algorithms can also be tested against the simpler conceptual vehicle models, which are available in the earlier stages of the program (Fig. 3.6b). These models are later

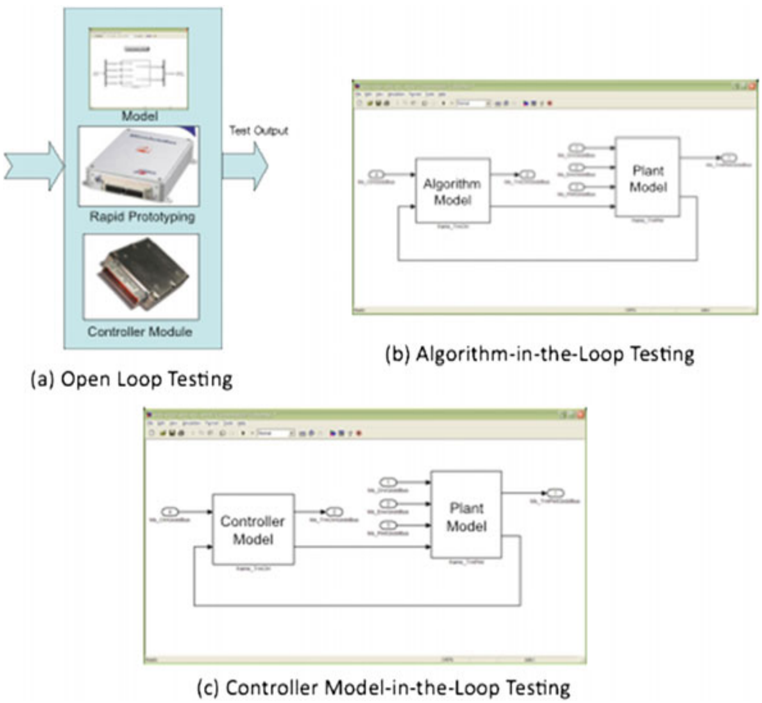


Fig. 3.6 Different types of testing in automotive industry

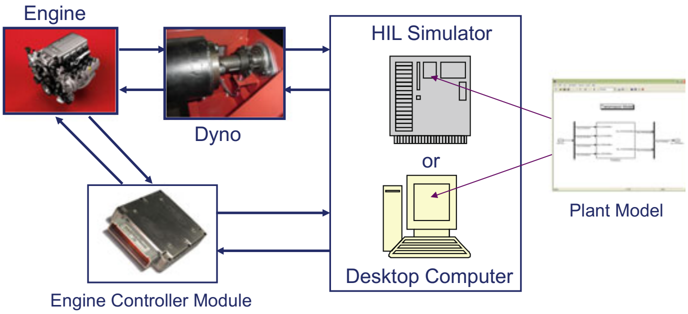


Fig. 3.7 Different types of testing in automotive industry (cont'd)

fortified with improvements based on component and vehicle testing data, which makes them suitable for more complex testing procedures such as module, component and vehicle in the loop types of testing.

In the later stages of the vehicle development process, a hardware-in-the-loop simulation can be run to see the proper operation of the vehicle controller using part real hardware and part simulations run in the computer environment as shown in Fig. 3.7.

3.1.2 Source of Models

Models are purposeful abstractions of the real world. With abstraction while certain aspects of the system are explicitly represented, other aspects are omitted that are not of concern (Topçu et al. 2016). They can be physical, mathematical, and/or logical (Sokolowski and Banks 2010). The scaled aircrafts that are used in wind tunnels are very good examples of physical models. When they are not physical, models are composed of a series of mathematical equations and/or logical expressions. These models can be physics-based, data-based, or hybrid (combined).

Physics-based models can be defined as the ones which are essentially mathematical and the governing equations are based on physical principles such as thermodynamics laws or Newton's law of motion.

The application of physics-based models in engineering domain is so common. Since early the days of engineering, Newton's law of motion has been used for modeling rigid bodies. Dynamics of machinery is an engineering field that deals with forces and moments and their effects on the motion. The theory of machines studies the relative motion of machine elements under the effects of external forces (Khurmi and Gupta 1976).

Modeling the mechanical behavior as a continuous mass is the topic of continuum mechanics. It is concerned with the stress in the continuous medium (solids, liquids or gases) and their deformation or flow (Malvern 1969). Continuous as an adjective is used to express the approximation that assumes the mass without gaps and empty spaces thereby representing the mathematical functions as well as their derivatives are continuous. This hypothetical medium is called continuum. The governing physical laws in this case are conservation of mass, momentum, and energy. These equations will be summarized in Sect. 3.2.6. The motion of viscous fluids is mostly computed by applying Navier–Stokes equations which encompasses time-dependent equations for conservation of mass, momentum, and energy. Euler equations are well-employed simplification of Navier–Stokes equations which neglects the effects of viscosity (Schetz and Fuhs 2013). Computational Fluid Dynamics (CFD) is the area of study which applies numerical methods like finite difference or finite volume to solve the approximations of these equations. Physical model of heat has also been built considering it as a fluid inside the matter. Heat equation is a partial differential equation that concerns the distribution of heat in material over time (Widder 1976). Solid mechanics deals with the behavior of solid materials under load. While elasticity is the study of body that retains its original state after releasing the load, plasticity governs the nonreversible deformation of solid. Euler–Bernoulli beam equation and plate theory are well-applied simplifications in modeling and simulation of elastic behavior. They both define the relations between the applied forces and the resulting deflections (Fung 1965). Finite Element Method (FEM) as will be discussed later in Sect. 3.2.3., is commonly employed for approximating partial differential equations within Navier–Stokes equations, heat equation and Euler–Bernoulli beam equation (Dhatt et al. 2012). It promotes using simple approximation of unknown variables to transform partial differential equating to algebraic equations.

Data-based models utilize the data that describes the particular aspects of the system that is subject to modeling. It is also named as empirical modeling since the model depends on empirical observations rather than mathematical equations (Sokolowski and Banks 2010). While the computing power as well as the optimized implementations of finite element analysis and computational fluid dynamics software getting better, engineering design optimization of complex systems like aircrafts or cars requires long lasting simulations which are sometimes unacceptable in practice (Wang and Shan 2007). Additionally, sometimes it is required to incorporate data from the real world into the simulation. Data-based models, or simply called metamodels approximate computation-intensive functions or real-world data to analytical models. The modeling process starts with data collection using sampling methods such as fractional refactoring, or Latin hypercube. Then the model is constructed is a particular method of choice. Polynomial equations, splines, Multivariate Adaptive Regression Splines (MARS), artificial neural networks are some of these methods. Model fitting is done with an appropriate approach like least squares or backpropagation.

Hybrid modeling combines previously mentioned two modeling paradigms. While a part of a physical process is approximated using data models, the rest of the physical process is modeled using equations that represent the law of physics. In modeling and simulation of air vehicles, it is a common practice to develop data models for the aerodynamics modeling, where the flight dynamics is modeled using Newton's laws of motion (Jategaonkar et al. 2004). The aerodynamics data may be collected from the flight experiments, wind tunnel tests or with CFD runs. Nowadays, the design of complex multidisciplinary systems such as aircrafts, automobiles and similar is carried out using hybrid models within a Multi-Disciplinary Design Optimization (MDO) framework (Martins and Lambe 2013). Such procedures allow designers to incorporate all relevant disciplines simultaneously. The optimum of the coupled problem is superior to the design found by optimally designing each module sequentially, since it can exploit the synergistic coupling between them. However, this concurrent consideration results in a much more complex problem. Therefore, systematic structuring, modeling, and approximation tools have to be employed within MDO, which has been applied with successfully to the design of many commercial products.

3.2 Simulation of Continuum

The term modeling refers to the development of a mathematical representation of a physical situation whereas simulation refers to the procedure of solving the equations that resulted from model development (Ashby 1996). With the development of mathematical models it was possible for scientists to integrate research into natural phenomena within their investigations. Analysis of these models was only possible via existing analytical or numerical methods which by then were only tackled for specific problems. Each of these methods in literature is known by the great scientist who developed them such as Euler, Newton, and Gauss.

Despite major contributions by various outstanding scientists, the main issues concerning the theoretical and physical understanding of the equations in continuum mechanics are still being worked on. Continuum mechanics has changed dramatically since the late nineteenth century, so that theoretical studies are now coined with numerical experimentation and simulation. Furthermore, progress in the computational speed and power allowed researchers to develop mathematical models for much more complex physical problems some of which will be discussed in the multi-scale and multi-physics sections. After the invention of calculus, many advanced PDE's were introduced to describe the physics of systems from different disciplines such as solid mechanics, fluid mechanics, and elastodynamics. Important contributions were initially made by Euler, Lagrange, and Cauchy and these were followed by the application of PDE's to describe the physics of electromagnetic (EM) theory by Maxwell, Heaviside, and Hertz, and finally to quantum mechanics with major theoretical work by Schrodinger. These equations are descriptive of the time evolution and relationship of various fields in a three dimensional space. Major efforts of continuum simulation in the areas of Solid Mechanics, Fluid Mechanics, and Electromagnetics will be described in the next sections to follow.

The introduction of efficient and powerful platforms enabled researchers to solve the constitutive laws of continuum in mechanics in combination with the laws of conservation of mass, energy, and momentum. The same is valid for other fields including EM. Some of the most popular methods used for this purpose are the Finite Element Method (FEM), Finite Volume Methods (FVM), Finite Difference Methods (FDM), and Boundary Element Methods (BEM). These methods are applied to the simulation of matter in all forms, i.e., solids, liquid, and gas, based on a major assumption of continuum media, thus Computational Mechanics of Continua. Namely, the term continuum describes the nonseparability of the considered domain and validity of continuity between any points in the domain so that differentiation is possible. Therefore, continuity between elements in any continuum-based numerical technique is maintained as well. Unlike analytical exact solutions of differential equations, which allow the solution at every point, the numerical solution is only calculated at chosen finite number of nodes, yielding in turn a reduction in complexity of the system. Well-known methods to conduct continuum-based simulation are described in the next section.

3.2.1 Finite-Difference Method

One of the earliest and widely used numerical method for solving PDE's within continuum mechanics is the Finite-Difference Method (FDM). The main idea of FDM is based on replacing the differential terms with respective to the spatial coordinates with the so-called finite differences over small enough distances based on the Taylor's series approximation. For that purpose, the domain of interest needs first to be discretizedinto vertical and horizontally located nodes, on which finite differences are defined. Several finite difference integration schemes exist known as forward, backward, and central difference schemes. It is worth noting that the FDM is equally applicable to time differentiations. As a result of discretizing the domain into nodes, a system of algebraic equations in terms unknowns at the chosen nodes are constructed. Each algebraic equation belonging to its corresponding node is expressed as a combination of function values at its own node and its neighboring nodes. Next step is to impose boundary conditions, which leads to the solution step of the equation system using either direct or iterative solution methods. Finally, unknowns at each node are solved. This solution is only an approximate solution since the finite differences are first-order approximations of the partial derivatives. The FDM when compared with the FEM or BEM allows for a direct discretization of the equations and does not rely on the use of interpolation functions. Therefore, it is one of the most direct and intuitive techniques that exist for the solution of PDEs. Moreover, for material nonlinearities, the FDM proves to be favorable as it allows their simulation without the need of iterative techniques. However, it suffers from relying on regular noded discretization scheme which makes modeling of irregular geometries a challenging task. This also results in difficulties when heterogeneous material compositions and unusual boundary conditions are present. However, the FDM has been generalized to overcome related shortcomings through methods based on irregular node/grid structures with methods such as irregular quadrilateral, triangular, and Voronoi grids.

3.2.2 Finite Volume Method

The Finite Volume Method is similar to the FDM method and evolved as its successor to solve PDE's with one major difference: these differential equations are expressed in integral form. Its formulation leads to the concept of finite volumes, which essentially correspond to volumes around and encompassing each node in a mesh. Similar to the FDM, algebraic equations of unknowns at nodes are built by replacing the integrals and by considering boundary and initial conditions. Thereby, the system of equations to be solved is constructed. The FVM, similar to the FDM has certain advantages such as allowing the usage of irregular unstructured mesh and modeling capabilities of nonhomogeneous material compositions.

3.2.3 Finite Element Method

The Finite Element Method (FEM) was introduced in the 1960s as an alternative method to FDM for the numerical solution of stress concentration. More importantly, it is the first numerical solution method which was capable of dealing with complexities such as nonlinearities, nonhomogeneous materials, complex geometries, and sophisticated boundary conditions. As a result, FEM was soon recognized as the most popular numerical method in continuum mechanics, mainly so because unlike FDM, it allowed for nonuniform discretization. The method was found more extensive and used a decade later with the theoretical developments made by Bathe (2006) and Zienkiewicz and Taylor (2005). Many researchers have contributed to the development of the method which is by far the most favorite method for the approximate solution of many sophisticated continuum mechanics problems of dynamic, anisotropic, and inelastic behavior. It is a generic numerical solution technique for boundary value problems coming from various disciplines. The main principle rests on the idea of dividing the problem domain into smaller subregions (areas or volumes) called finite elements. This is followed by typical steps of defining local element approximations, performing assembly of finite elements and ultimately solving the resulting global matrix equation. More specifically, the unknown function (e.g., displacement field, temperature field, electric field, velocity and pressure fields) is approximated via trial/interpolation functions of the nodal values (or edge unknowns in EM problems) using polynomial functions. Numerical integration is performed in each element using Gauss quadrature points. After assembly, the algebraic global system of equations is obtained. Because of continuum assumptions, standard FEM methods cannot be directly and efficiently applied to discontinuum problems involving cracks, damage-induced discontinuities or singularities and failure analysis.

In addition to the well-known superiority of the FEM which is well suited for complex analysis of systems composed of heterogeneous materials and irregular geometries owing to the possibility of using an irregular mesh, it also proved to be an appropriate tool for modeling various nonlinear geometries and inelastic material behavior and nowadays material hardening and softening. Moreover, it has the capability of representing geometric nonlinearities, contact mechanisms, fluid-structure interaction, multi-scales, etc., as will be discussed in separate sections below. Therefore, the FEM will stand out as the mostly used and diverse numerical method in continuum mechanics.

3.2.4 Meshless Methods

The bottleneck in applying FEM to complex engineering problems with intricate geometries, unusual material properties, and complex boundary conditions is the mesh generation process, which usually in 3D problems is an extremely challenging task mostly comparable to the problem solution itself. Another disadvantage of FEM relates to numerical instability due to a distorted mesh. Both of these problems can be avoided by another class of methods known as 'meshless methods', which as the name implies does not rely on elements but interpolation functions are generated from neighboring nodes within a domain of influence. More specifically, nodes are created across the domain without the need of a fixed element topology definition. As a result, the interpolation functions obtained are no longer polynomial functions leading to more difficult numerical integration when compared with the FEM where Gauss integration points are used. Moreover, meshless methods suffer from increased computational requirements but do not rely on standard mesh generators and are able to easily represent more complicated geometries. In literature, methods such as smoothed particle hydrodynamics, diffuse element method (DEM), element-free Galerkin method, reproducing kernel particle methods, moving least squares reproducing kernel method, hp-cloud method, the method of finite spheres, and finite point method stand out.

3.2.5 Multi-scale Methods

All products whether man-made or natural are composed of multiple scales. Taking an example from the aeronautical industry, the Airbus A380 consists of many thousands of structural components and many more sub-structural details. Similarly, its fuselage consists of 750,000 holes and cutouts with different structural and material scales. When viewed at the roughest material scale, fuselage composites' part consists of woven/textile composite and laminate scales; at the intermediate scale, it is composed of a tow or yarn, which consists of a bundle of fibers. When looked at a more discrete scale, including atomistic and ab initio scales, the aircraft's metal part consists of a polycrystalline scale, a single crystal scale, a discrete dislocation scale, and also time atomistic and ab initio scales.

Computations and simulations in the aforementioned multi-scales have been identified as areas of utmost importance to advance the future in nanotechnology. One of the obvious fundamental challenges associated with such a multi-scale approach relates to the increased uncertainty and complexity introduced by these finer scales. However, the application of any multi-scale approach has to be carefully evaluated. For instance, considering metal matrix composites with fibers arranged in a periodic fashion, finer scales could prove useful because the bulk material typically does not obey normality rules, and the development of a phenomenological coarse-scale constitutive model would be extremely difficult. This would also allow a better understanding of each phase and the overall material response could be extracted from its fine-scale constituents via homogenization techniques. However, for brittle ceramic matrix composites, with microcracks that exist in a random distribution and complex interface properties, are difficult to characterize, a multi-scale approach would not be an appropriate alternative.

There are two main categories of multi-scale approaches in literature, namely, hierarchical or concurrent. In the former approach, the fine-scale response is idealized/approximated and its overall/average response is integrated into the coarse scale. In the latter approach, fine and coarse-scale resolutions are simultaneously employed in different portions of the same problem domain, and the exchange of information occurs through the interface. The sub-domains which present themselves at different scale resolutions can be either overlapping or disjoint.

Various hierarchical multi-scale methods have been labeled by different names, including upscaling, coarse-graining, homogenization, or simply multi-scale methods. There are also subcategories of the above definitions, such as systematic upscaling, operator upscaling, variational multi-scale, computational homogenization, multigrid homogenization, numerical homogenization, numerical upscaling, and computational coarse-graining, just to mention a few. Moreover, different definitions are used to indicate various scales. If the structure exists in two scales, however, the fine scale is often referred to as a micro-scale, unresolvable scale, atomistic scale, or discrete scale; the coarse scale is often defined as the macroscale, resolvable scale, component scale, or continuum scale. For more than two scales, the additional scales may be termed as mesoscales.

One alternative approach to the homogenization of artificial structures avoiding the limitations associated with earlier analytical homogenization models (Milton 2002) is the theory of a mathematical homogenization approach. It is based on the asymptotic expansion also known as two-scale homogenization, which is a well established concept in the theory of PDEs with rapidly oscillating periodic coefficients (Bensoussan et al. 1978). Its main advantage is that the method is generalized enough and unlike analytical techniques, can handle unit cells with inclusions of arbitrary geometry and any number of phases with no additional computational cost. Also, instead of formulating the problem as an eigenvalue problem, two-scale homogenization works directly on the original form of the governing equations and is therefore able to result in expressions valid for effective constitutive tensors.

In their studies, (El-Kahlout and Kiziltas 2011) further developed this approach by applying two-scale homogenization method to Maxwell's equations and extracting the effective parameters of periodic dielectric and magnetic materials, that can be in their most generalized form lossy and are made of inclusions with arbitrary shapes and multi-phase material constituents. The numerical solution of the resulting PDE is carried out using a commercial FEA based solver, namely COMSOL Multiphysics, where the effective tensors are evaluated at a single frequency, for both isotropic and anisotropic effective material tensors with isotropic constituents. This is the first study where numerical material model based on two-scale homogenization is used to synthesize the microstructure of EM material with desired material matrices using formal design techniques such as topology optimization. Results of this design study are shown in Fig. 3.8 which was also fabricated (El-Kahlout and Kiziltas 2011) using novel Dry Powder Deposition techniques as demonstrated in Fig. 3.9.

Similar to EM materials, most heterogeneous materials, such as composites, polycrystals, and soils consist of constituents/phases with clear-cut boundaries that display different mechanical and transport properties. The use of homogenization of continuum allows a better understanding of the physical governing equations of individual phases, including their geometry and constitutive equations at the fine-scale phases, or at least a better grasp than at the coarse-scale phases. Put in other way, the process of homogenization provides a mathematical means by which coarse-scale equations can be deduced from well-defined fine-scale equations. Moreover it allows the determination of heterogeneous material behavior, at least theoretically without the need of testing, which is usually a very expensive endeavor. Also, through homogenization, one can estimate the full multiaxial properties and responses of heterogeneous materials, which present themselves as anisotropic materials and are most of the time extremely difficult to measure experimentally. In addition to describing the overall behavior of heterogeneous materials, the act of homogenization leads to local fields via the process known as downscaling given coarse-scale fields, phase properties, and phase geometries. This information is of critical importance in understanding and describing material damage and failure.

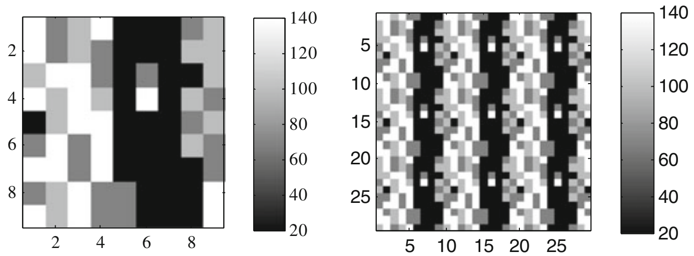


Fig. 3.8 Optimal material distribution (dielectric ranges from 20 to 140) of designed unit cell (*left*) and array (*right*) for a desired permittivity tensor of ε = [45.0;’0.70] using mathematical homogenization and topology optimization.[reproduced courtesy of The Electromagnetics Academy]

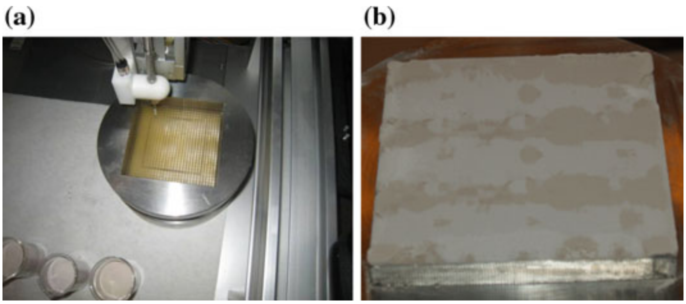


Fig. 3.9 Automated fabrication of design in Fig. 3.8 using dispensing machine within DPD in action (*left*) and resulting desired deposited substrate (*right*). [Reproduced courtesy of The Electromagnetics Academy]

3.2.6 Solid Mechanics

Two main branches exist within the application of the principles of mechanics to bulk matter: the mechanics of solids and fluids. When viewed from a global perspective, the common subject is that of continuum mechanics. More specifically, continuum mechanics conceives the useful model of matter as continuously divisible, and does not make any reference to its discrete structure at microscale, which is well below those scales of the phenomenon of interest. Solid mechanics is concerned with stresses, deformation, and failure of structures and solid matter. A material is called a solid and not a fluid if it is able to support significant amount of shear force over a certain time period of a natural process or technological application of interest.

The main equations of continuum physics can be presented by separating them into global and local laws. The former serve as the foundations of continuous media theory and are summarized here (Muntean 2015). In all these formulations, $\Omega'(t)$ denotes arbitrary configuration of partial volume B′ of B. More specifically, global balance laws for the five major conservation principles are presented here for mass, linear and angular momentum, energy, and entropy.

Mass:

The conservation of mass is expressed in its most general form as

$$\frac{d}{\mathrm{dt}} m(\Omega'(t), \mathbf{t}) = 0 \tag{3.1}$$

for all $\Omega'(t) \subset \Omega(t)$, where $m(t)$ stands for the total mass in $\Omega'(t)$, i.e.,

$$m(\Omega'(t), \mathbf{t}) = \int_{\Omega'(t)} d\mu_{\mathrm{m}} = \int_{\Omega'(t)} \rho dx \tag{3.2}$$

with $\rho(\mathbf{t}, x)$ denoting the density. Assuming that there is no internal mass production, (3.2) states that the total mass of any material partial volume is conserved.

Linear Momentum:

The conservation of linear momentum or balance of forces is expressed in its most general form as: For every part $\Omega'(t) \subset \Omega(t)$ we have

$$\frac{d}{dt}\ell(\Omega'(t), \mathbf{t}) = \mathbf{F} \tag{3.3}$$

where the linear momentum can be defined via

$$\ell(\Omega'(t), \mathbf{t}) = \int_{\Omega'(t)} \mathbf{v}\,\rho dx \tag{3.4}$$

More specifically, the time rate of change of total linear momentum of $\boldsymbol{l}$ in B′ is equal to the force $\mathbf{F}$ exerted on B′. The force $\mathbf{F}$ consists of the contribution of the internal body forces per unit of volume $\rho\mathbf{f}_b$ and contact or surface forces per unit of area $\mathbf{t}$ acting on the boundary ∂B′ of B′. Here, $\mathbf{t}$ is the stress vector or traction.

Angular Momentum and Moment of Momentum:

The conservation of angular momentum or balance of moments is expressed in its most general form as

For every part $\Omega'(t) \subset \Omega(t)$, we have

$$\frac{d}{\mathrm{dt}}\boldsymbol{\alpha}(\Omega'(t), \mathbf{t}) = \mathbf{M} \tag{3.5}$$

where the angular momentum can be defined via

$$\boldsymbol{\alpha}(\Omega'(t), \mathbf{t}) = \int_{\Omega'(t)} \boldsymbol{x} \times \boldsymbol{v}\;\rho\,dx \tag{3.6}$$

More specifically, the time rate of change of total angular momentum $\boldsymbol{\alpha}$ of B′ is equal to the moment $\mathbf{M}$ of the force $\mathbf{F}$ exerted on B′.

Energy:

The conservation of energy balance is expressed in its most general form as: The time rate of change of the total energy within B′ which is composed of the kinetic energy K and internal energy E and is equal to the rate of work, say P, done by both the body force and the contact force, plus the heat supply Q from internal heat production and heat fluxes across the boundary of B′. So for every part $\Omega'(t) \subset$ (t), this can be written as

$$\frac{d}{\mathrm{dt}}(K(t)+E(t)) = P(t)+Q(t) \tag{3.7}$$

where

$$K(t) = \int\limits_{\Omega'(t)} \frac{|\mathbf{v}|^2}{2} d\mu_{\mathrm{m}} = \int\limits_{\Omega'(t)} \frac{|\mathbf{v}|^2}{2} \rho\, dx, \tag{3.8}$$

$$E(t) = \int\limits_{\Omega'(t)} e\, d\mu_m = \int\limits_{\Omega'(t)} e\, \rho\, dx, \tag{3.9}$$

$$P(t) = \int\limits_{\Omega'(t)} \mathbf{v} \cdot (\mathbb{T}\mathbf{n}) d\sigma = \int\limits_{\Omega'(t)} \vec{f} \cdot \vec{v}\, \rho dx, \tag{3.10}$$

$$Q(t) = \int\limits_{\Omega'(t)} f_{\mathrm{Heat}} \rho\, dx + \int\limits_{\Omega'(t)} \mathbf{q} \cdot \mathbf{n} d\sigma. \tag{3.11}$$

In Eq. (3.9), e represents the inner energy density. The first term in $Q(t)$ accounts for the heat source. The measure μ_{m} in the equations of $K(t)$ and $E(t)$ corresponds to the mass measure associated with the material body B.

Entropy:
The entropy increase within B is greater than or equal to the internal entropy supply, i.e., internal heat source over θ, which is the absolute temperature, plus the entropy flux across the boundary of B′, which can be expressed as follows:

$$\frac{d}{\mathrm{dt}}\left(\int\limits_{\Omega'(t)} s\, d\mu_{\mathrm{m}}\right) \geq \int\limits_{\Omega'(t)} \frac{f_{\mathrm{Heat}}}{\theta} d\mu_{\mathrm{m}} - \int\limits_{\partial\Omega'(t)} \frac{\mathbf{q} \cdot \mathbf{n}}{\theta} d\sigma. \tag{3.12}$$

Here s represents the entropy density. It is explicitly noted that all conservation laws are in term of extensive quantities. More specifically, global balance laws can only be written in terms of extensive quantities. However, the intensive quantities are related to local balance laws expressed in terms of PDEs and inequalities as well as boundary conditions and can be derived based on global laws of the preceding section (Muntean 2015).

3.2.7 Fluid Mechanics

Various theories govern the physics of fluid mechanics and different methods are proposed and used in literature to provide numerical solutions/simulations primarily depending on the spatial and temporal scale of the phenomenon. Instead of going

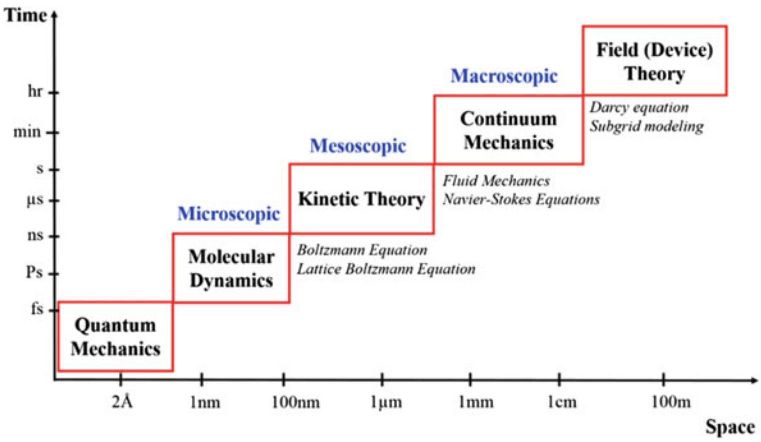


Fig. 3.10 Typical numerical methods used in fluid mechanics based on temporal and spatial scale

into detail with all methods, these theories and typical numerical methods employed according to the temporal and spatial scales are summarized in Fig. 3.10. As depicted in the graph, continuum mechanics prevails for above microscale and below tens of meters with a time scale between 1 s and hours. When the continuum assumption breaks down, the fluid has to be described by an atomistic point of view, such as the molecular dynamics as a microscale method or statistical rules govern the molecular group behavior, i.e., kinetic theories as mesoscopic methods for larger scales. On the spatial and time scale limit, if the characteristic length is smaller than 1 nm or the characteristic time is shorter than 1 fs, the quantum effect may not be negligible for the system of interest and quantum mechanics has to be brought into describe the transport. In fact, modeling at a smaller scale may present a more accurate description of the problem, but is likely to cause a much higher computational cost. Therefore, as always in numerical simulations, in engineering an appropriate tradeoff is considered when trying to determine in an accurate and fast way, the fluid behavior of interest.

Despite the emergence of high-speed platforms and the advances in efficient and accurate numerical methods, some computational fluid dynamics (CFD) problems still present themselves as challenging problems for the practical solution via numerical simulation techniques. For example, NASA has recently modified its aerospace design codes for earth science applications, thereby speeding up supercomputer simulations of hurricane formation (Kazachkov and Kalion 2002). An example of such a CFD simulation using a 512-processor supercomputer is referred to in (Kazachkov and Kalion 2002). More specifically, actual data from a variety of different sources and climate models were integrated to generate high fidelity simulations so as to reproduce a hurricane forming in the Gulf of Mexico. As a result, engineers were able to simulate the formation and movement of a hurricane. However, the weather forecast of global earth based on CFD atmospheric and ocean simulation is still a challenging problem, and calls for even larger amount of computing power and more accurate data. Overall, this is a multi-phase CFD problem with very complex geometry and dynamic boundary conditions.

3.2.8 Electromagnetics

In electromagnetics (EM), matrix systems with a few millions of unknowns known as dense matrix systems have been solved numerically for ten years now. Today, the number of unknowns that can be solved via simulations is on the order of a billion of unknowns (Gurel and Ergul 2007). This impressive improvement is attributed to the synergistic progress between hardware and algorithm design. It is also noted that for sparse matrix systems resulting from specific simple EM problems within electrostatic and magnetostatics, even larger scale problems can be addressed such as the World-record algorithm from Jülich calculating over three trillion particles (World-Record Algorithm from Jülich Calculates Over Three Trillion Particles—Research in Germany 2011).

As stated earlier, the physical response of many fields including EM, can be analyzed via differential equations. Therefore, PDE's were used for more than four centuries and continue to set the standard for modeling the physics of different media today. There are three main groups of differential equations, namely hyperbolic, parabolic, and elliptic, which describe fields with various physics. The Laplace equation or the Poisson equation given below is a well-known example of a generalized simple elliptic PDE. These are encountered in the numerical modeling of EM problems in the static regime and various transport problems. They are known for characterizing fields or potentials associated with no singularities distant from the source location, or equivalently these equations are differentiable functions and therefore do not allow for any singularity propagation.

$$\nabla^2 \varphi(\mathbf{r}) = -\frac{\rho(\mathbf{r})}{\varepsilon}. \tag{3.13}$$

Typical examples of parabolic equations, the second group of PDE's, are the Schrodinger and diffusion equations. These equations are characterized by their first time derivative and second space derivative. These are fundamental equations in quantum mechanics and heat transfer as well as low-frequency EM propagation in conductive media, respectively. A diffusion equation in standard form is

$$\nabla^2 \varphi(\mathbf{r}) - \frac{1}{c\,\tau}\frac{\partial}{\partial t}\varphi(\mathbf{r}) = 0. \tag{3.14}$$

The third class PDE's refers to hyperbolic equations, and an example belonging to this group is the wave equation. It has second-order space and time derivatives.

$$\nabla^2\varphi(\mathbf{r}) - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\varphi(\mathbf{r}) = 0. \tag{3.15}$$

Solution of differential equations as earlier denoted are carried out by three major methods: a subspace projection method (e.g., FEM), the FDM, and the pseudo-spectral method. Various basis/interpolation functions are introduced to fit the unknown field (Chew 1995) in the subspace projection method. It covers a subspace of the larger space that the field is defined over due to the finite characteristics of basis functions. Thereby, the PDE is easily converted to a time dependent ordinary differential equation. For the solution of the equation via time stepping or marching, the derivatives can further be approximated using finite difference or the subspace projection method. Or as an alternative, time domain Fourier transform can be used to remove the time derivatives resulting in a matrix equation to be solved via iterative or inversion techniques.

A major alternative exists to the numerical solution of the governing Maxwell's equations in EM expressed in PDE. Specifically, initially a point source response called Green's function can be introduced. Based on linear superposition and as a result of an arbitrarily distributed source, the unknown field is obtained via spatial convolution of the distributed source expressed via Green's function. This corresponds to the equivalence principle (Harrington 2001) which allows the field in a given region to be expressed as Green's operator acting on the sources. Hence, the resulting equations are of integral equation type (IE). When compared with PDE, IE have an important advantage where the EM unknowns correspond to only surface unknowns, or to volume unknowns that occupy only a spatial finite region. Therefore, the number of unknowns in the IE formulation may be much less than those in the PDE formulation. More importantly, this IE formulation leads to the automatic satisfaction of the radiation condition if a suitable Green's function is chosen. However, in the PDE formulation absorbing boundary conditions or the so-known boundary integral equations replace the radiation condition. Additionally, using the subspace projection method (Harrington 2001), these IE can be converted into matrix equations. Equivalently, operators of the integral are replaced with matrix operators. However, the matrix representation of Green's operator corresponds to a matrix system which is dense because of its non-local nature. Hence, the computational storage and operations such as matrix vector products with that type of a matrix system can be computationally expensive. In literature, some methods have been developed to overcome these expensive matrix solutions. These include fast Fourier transform based methods, fast-multipole-based methods, rank-reduction methods, the nested equivalence principle algorithm, recursive algorithms, etc. (Weng et al. 2001).

As a final class of numerical techniques for EM radiation and scattering problems, hybridized versions of the two main classes combining their advantages have been developed. The FE-Boundary Integral (BI) method is one of the most powerful techniques belonging to this class. More specifically, it offers the flexibility of the FEM to analyze structures with highly complex geometrical and material details

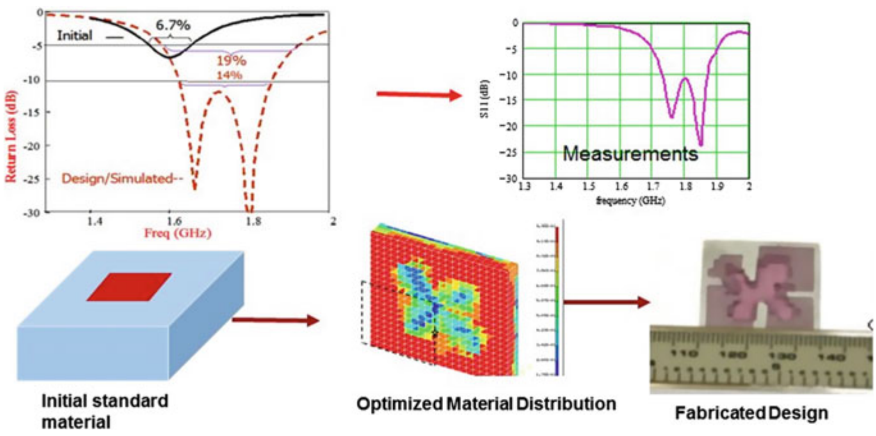


Fig. 3.11 Design results of novel material distributions of a patch antenna via integration of FE-BI method and topology optimization (Kiziltas et al. 2003)

but at the same time imposes a rigorous boundary condition via the use of the BI formulation. This tool's efficient and accurate analysis capability has allowed researchers to conduct numerous designs (Volakis et al. 2006).

It is especially noted that these efficient and accurate codes allowed for the first metamaterial-based antenna design using topology optimization based techniques as shown in Fig. 3.11 (Kiziltas et al. 2003). The design developed from scratch as shown in Fig. 3.11 was based on 5 individually textured layers which were also fabricated and measured. The agreement between measurements and calculations is truly impressive for the complex dielectric design. Above all, the threefold improvement in bandwidth is a clear demonstration of the remarkable potential of efficient and accurate numerical techniques in delivering novel designs not only in EM but also in other engineering disciplines.

Multi-scale problems as discussed in Sect. 3.2.5, present themselves in circuits, packages, and chips at various levels of complexity. Similarly, they exist also in antennas on complex platforms, in nano-optics and nanolithography applications. Therefore, multi-scale solutions of problems are critical for many applications. Similar to other applications, the size evaluation of the EM multi-scale problem is of great importance. More specifically, one needs to evaluate the multi-scale structures relative to the wavelength to determine which physics of the three to apply for their solution: circuit physics, wave physics, or optics physics. Avoidance or identification of ill-conditioned numerical systems plays a great role in the effective solution of multi-scale EM problems.

It is finally noted, that one of the biggest challenges today in the numerical solutions of EM problems is the model size of realistic problems which deems high-performance computing a vital necessity. Significant speedups have been achieved by hardware scaling and additional efforts have resulted in three main types of HPC platforms: (1) supercomputers, (2) computer clusters, and (3) cloud computing. It is quite evident that, computational EM and large-scale computing will continue to evolve given that these are indispensable tools for EM analysis and design. Not only will it allow for efficient and practical performance evaluation and novel designs but it is expected to continue the enhancement of our thorough understanding of the physics within highly complex systems.

3.2.9 Multi-physics Methods

Many realistic problems present themselves as very complex problems due to their multi-physics nature. Scientists and engineers from various fields have been working on the combination of different numerical techniques with the goal of addressing these elaborate physical processes, such as the transition from continuum to discontinuum (e.g., fracture processes) or the interaction of multi-phases of matter (e.g., hydrofracture processes). As a result, a new class of numerical methods called hybrid/multi-physics methods evolved. It is due the developments in high-performance computing and computational science and computer hardware that this group of methods evolved. Major examples are: Combined Finite-Discrete Element Method (F-DEM), Hybrid Lattice Boltzmann-FEM, Lattice Boltzmann-DEM, etc. Areas of interest include algorithms and novel solutions for:

- Coupling of FEM and DEM simulations
- Coupling of FEM and/or DEM with CFD solvers
- Coupling of different solvers of continuum mechanics, e.g., FEM-FVM.
- Coupling of continuum and discontinuum mechanics solvers, e.g., FEM-DEM, FEM-MPM, FEM-LBM, etc.
- Coupling of solid and fluid mechanics solvers, e.g., FEM-LBM, FEM-FVM, etc.
- Coupling of discontinuum mechanics solvers, e.g., DEM-SPH, DEM-LBM, etc.
- Coupling of solvers for different scales, e.g., coupling of FEM-DEM.

3.3 Simulation of Machinery

In many simulation studies developers represent the components of the target system based on their dominant energy-based properties. Although various linear and nonlinear extensions exist, basic energy-based properties can fundamentally be given as inertia, storage (spring), and dissipation (damping). For example, in many engineering simulations a bearing is represented as a damping element and an axis slider in a manufacturing system is represented as inertia. It is also noted that a bearing component brings a negligible rotational inertia to the system as well as a slider which is somewhat flexible and its dimensions may change very slightly under heavy operating conditions. But these are not considered as dominant properties for these components.

The approach of representing complex and spatially distributed physical systems based on their dominant energy properties is known as the "lumped parameter modeling", implying the dominant energy-based characteristic(s) of a component are represented by using specific and predetermined elements (Karnopp et al. 2000). The use of lumped parameter systems approach results in a more structured approach of developing simulations for complex engineering systems. Using energy rather than other physical features (force, current, etc.) also makes it possible to use this approach in multi-domain systems.

3.3.1 *Single Degree of Freedom Systems*

Generally, in engineering, the classification of lumped parameter systems is given based on the number of the inertia elements. In many cases especially for mechanical systems, the freedom of motion of the component represented as inertia is important. For example, if a component can move in both x and y axis and/or can also rotate about z axis, these motion properties are all represented as separate inertial elements and flow variables.

Single degree of freedom systems are systems represented with one inertial element and have one variable governed by fundamental physical equations. A good example for a single degree of freedom system is the longitudinal motion simulation of vehicles as shown in Fig. 3.12.

The system represented in this figure is given in (3.17) as a mathematical relationship (i.e., model) based on Newton's second law, where wheel traction forces, F, are the input and the vehicle acceleration, a_x, is the output.

$$ma_x = (W/g)a_x = F_{\mathrm{xr}} + F_{\mathrm{xf}} - W\sin\Theta - R_{\mathrm{xr}} - R_{\mathrm{xf}} - D_A + R_{\mathrm{hx}} \tag{3.17}$$

This mathematical model is straightforward to apply in simulation environments such as MATLAB/Simulink. The single degree of freedom longitudinal simulation can be used in basic fuel economy and traction (acceleration/braking) studies as

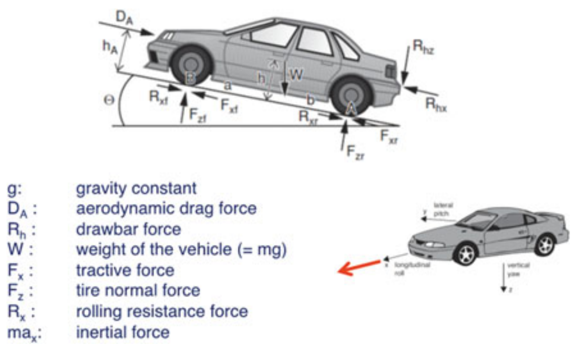


Fig. 3.12 Longitudinal motion of a vehicle

reported in Rajamani et al. (2000), Ulsoy et al. (2012). However for many vehicle engineering studies such as axle-based traction control (Cakmakci et al. 2011; Dokuyucu and Cakmakci 2016) more complicated representations (i.e., higher fidelity simulations) that are also suitable for V-process development model discussed in Sect. 3.1 is needed.

3.3.2 Multi-Degree of Freedom Systems

One way to improve the fidelity of the simulations is to increase the degree of freedom of its underlying mathematical model. This can be done by increasing the number of flow variables representing the inertia element, or adding more inertia elements to the system simulation. As an example of increasing the fidelity of the model by adding new flow variables to the inertia representing is the half-car model for vertical motion given in Fig. 3.13.

In Fig. 3.13, the vertical motion of a vehicle is represented with two degrees of freedom (translation and rotation about the center of mass) rather than only the vertical motion of the mass of the vehicle. A detailed mathematical model describing this system can be found in ("Automotive Suspension—MATLAB Simulink Example," n.d.) 2017, using road elevations, q, as input and vertical movement of the center of mass, z, and body rotation, θ as outputs. With this representation in simulations, the vertical motion of the occupant area can be studied as well as the wheel based vertical road force, which is critical for traction control studies such as wheel-based braking, and acceleration with so-called load transfer.

Another way of increasing the content is to increase the number of inertial elements considered in simulations. In this case, rather than using a single system boundary, where all of the components were lumped together before, can be broken into components and their relative interaction can be studied.

A good example for this kind of situation is the quarter car model shown in Fig. 3.14. In this model, a quarter of vehicle vertical dynamics is studied using quarter of the mass of the vehicle with the suspension system represented by k_s and c_s, f, tire parameters, k_{us} and c_{us} and vertical motion variables, z.

The mathematical equations representing this simulation is given in (3.18) based on Newton's second law:

$$\begin{gathered} m_s\ddot{z}_s + c_s(\dot{z}_s - \dot{z}_{us}) + k_s(z_s - z_{us}) = -f \\ m_{us}\ddot{z}_{us} + c_s(\dot{z}_{us} - \dot{z}_s) + k_s(z_{us} - z_s) + c_{us}(\dot{z}_{us} - \dot{z}_0) + k_{us}(z_{us} - z_0) = f \end{gathered} \tag{3.18}$$

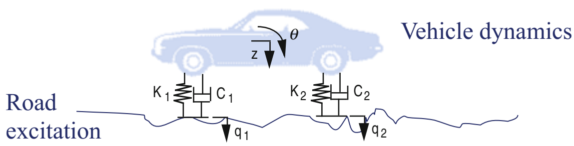


Fig. 3.13 Vehicle vertical dynamics

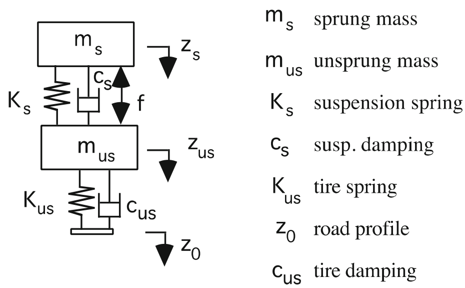


Fig. 3.14 Quarter car model for vertical motion simulations

It is important to note that by adding a new inertial element representing the mass of the wheel hub and suspension frame, the stroke motion of the suspension can be studied including the effects from the tire and the vehicle inertia which is not possible with the system given previously in Fig. 3.13.

The quarter car model given in (3.18) is an example of a multicomponent simulation in single axis of motion. More complicated models can be used to study wheel based multi degree of motion as given in Figs. 3.15 and 3.16 respectively for vehicle vertical motion (Rajamani and Hedrick 1995) and full vehicle motion.

In engineering simulation studies, as the number of components and the elements representing these components increase, the number of mathematical equations representing used in these simulations also increase. Therefore, the appropriate simulation content should be chosen to do the correct analysis with optimal computation time. For example, the full car model given in Fig. 3.16 is executed by solving 18 nonlinear equations per simulation step size as compared to the longitudinal model given in Fig. 3.12 contains only one ordinary differential equation. Both of the models can be used for fuel economy studies and both of them will contain uncertainties in inertia, spring and storage parameters.

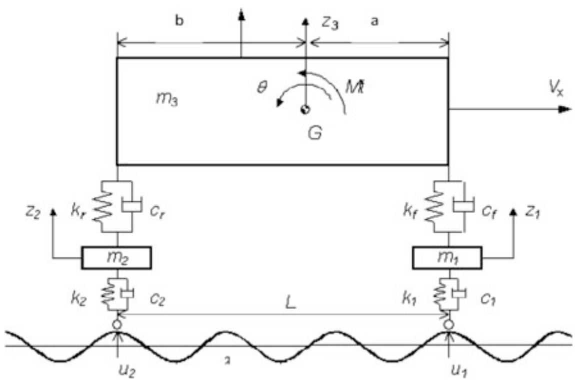


Fig. 3.15 Standard half-car model

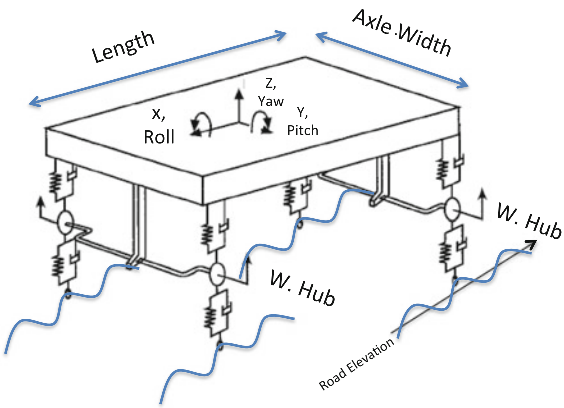


Fig. 3.16 Full car (18DOF) model

3.4 Simulation of Multi-domain Systems

In modern engineering systems, components that operate primarily in different domains (such as mechanical, electrical, digital) work together to complete complicated tasks. Therefore, a realistic simulation of the system should include the elements from different domains such as mechanical parts, power electronics to energize these mechanical parts, and digital components to monitor and/or control the operation.

Two representative cases of multi-domain simulations improving the performance can be given as the multi-domain simulations of electromechanical systems with controllers and online simulations to improve smart mechatronic/robotic systems.

3.4.1 Control Systems

Generally, algorithms in control systems are designed to have a certain dynamic behavior which can be represented in terms of a transfer function (in Laplace Domain) or a state space model (Ogata 1990). When these algorithms are actually implemented in actual systems their performance shows variations (usually degradations) due to the effect of execution in digital medium. These variations are due to the digitization of the algorithm, lack of realistic representation of the control system hardware and the effect of a communication in common medium such as networks.

Many control algorithms are developed by using a frequency or a time domain-based structured method based on their dynamic properties as discussed in many sources in control literature (Chen 1995; Ogata 1995). Once the development is finished, the resulting output is a fractional function that represents the input/output relationship of the control algorithm called controller transfer function, C(s), where s is the Laplace variable. The dynamic controller relationship can also be represented by a matrix equation pair generally given in the form $\dot{x} = Ax + Bu, q = Cx + Du$ where u is the controller input, q is the controller input, and x is the controller states. These representations both imply a continuous system where calculations or events take place instantaneously using a specific order. However, when implemented in a real-time control system, algorithm computations take certain amount of time to finish before an updated command can be issued. For many systems with fast dynamics, the effect of implementation generates a deficiency in performance since the optimal performance was designed for a medium where events take place instantaneously.

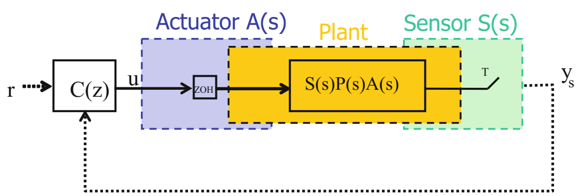
More realistic and predictable results can be obtained by using discrete control systems that take into consideration of the digital timing in their formulation (Franklin et al. 2009; Ogata 1995). Algorithms can be designed and simulated as digital controllers using adaptations of the continuous methods. Alternatively, continuous controller functions can be digitized afterwards using simple methods. For example by using a direct conversion approach, a controller transfer function C (s) can be converted to its discrete version by replacing the s operator with $(z-1)/Tz$ using a backward difference transformation. In this recipe z is the discrete variable and T is the sampling period.

Another important aspect of implementation of an algorithm in the digital world is the effect of *quantization*. Real numbers can have infinitely large digits during calculations, however for computers, it is more practical and maintainable to do operations in chunks of bits causing the calculations to take place in limited digits, which cause round off errors [Franklin et al.]. The effects of digitization and quantization can both be included in simulations to predict possible controller performance degradation in engineering systems.

Another important aspect in implementation of the control systemsis to include the hardware related properties such as un-modeled sensor/actuator dynamics (S (s) and $A(s)$ respectively) and the effect of sampling as shown in Fig. 3.17.

In many controller development activities, the controller is designed based on the plant dynamics $P(s)$ only without including the sensor ($S(s)$) and actuator ($A(s)$) dynamics. The dynamic response effects provided by actuators and sensors can be included in simulations by using time delays, noise, and offsets. The effect of digital to analog conversion in the actuator is modeled as a zero-order hold (ZOH) element that keeps the value of the actuator output constant for one time step. This element also represents the fact that the actuator has internal dynamics and cannot change its output instantaneously. A sampler element is used at the sensor to represent the

Fig. 3.17 Feedback system with device boundaries and sampling



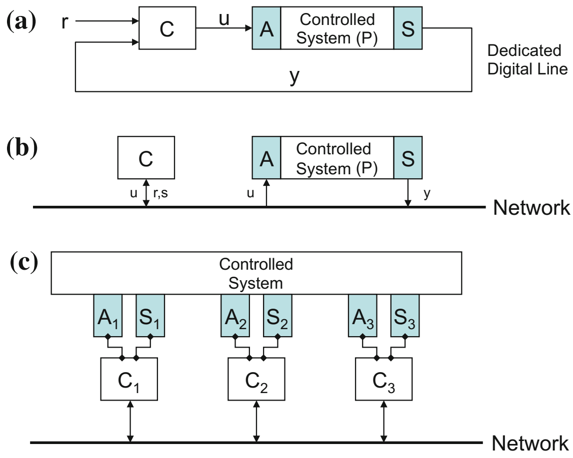


Fig. 3.18 Dedicated digital communications (**a**) versus networks (**b**, **c**)

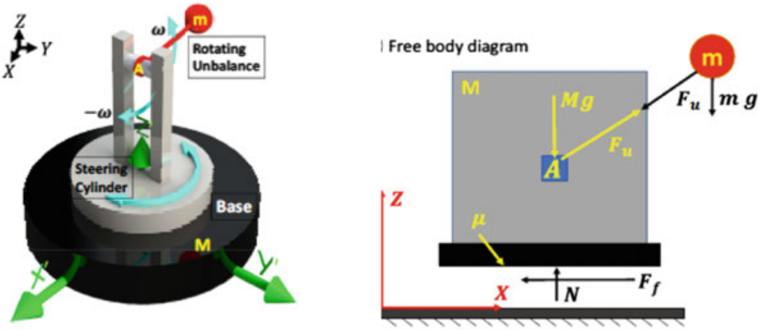
analog to digital sampling with rate T. This models the behavior of the sensor that it can only report plant outputs in every T seconds. Adding these effects to the overall simulation of the system provides more realistic performance studies.

Finally, in today's engineering applications, a common approach is to use communication networks instead of dedicated digital communication lines as shown in Fig. 3.18a, b. In fact, the benefit of this networked structure is being able to integrate as many components together with the capability of increased resources and easy maintenance as shown in Fig. 3.18c (Cakmakci and Ulsoy 2009). However, with the introduction of networks, the communication among system components can experience delays (or even loss of contact) as reported and studied by many researchers (Lian et al. 2002; Walsh et al. 2002). To remedy this effect, the overall system can be simulated using worst communication delays possible to measure the performance using step size-based delay elements and the controllers are calibrated accordingly.

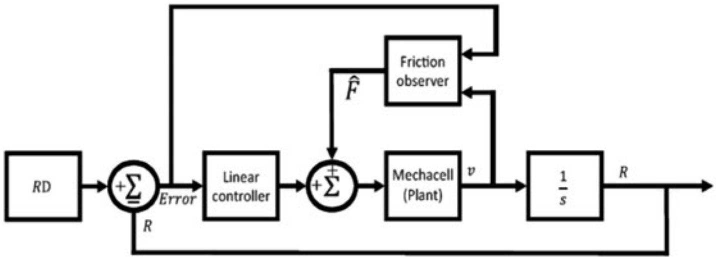
3.4.2 *Robotics and Cyber-Physical Systems*

One of the important use of simulations that predict system performance after the product design phase is to employ them as observers and/or monitoring threads in actual systems running in parallel and making predictions/modifications to improve system performance.

A good example of this type of utilization is the *friction observers* in robotic locomotion devices such as the one developed in Ristevski and Cakmakci (2015) as shown in Fig. 3.19. Many non-wheeled robotic systems observe the friction force during translation. Inside the controller, a simulation of the whole system based on the dynamic force balance is run to predict the effective friction force called the friction observer. The friction predictions from this observer is used to update level



(a) Vibration based translational system



(b) Translation Controller using a Friction Observer

Fig. 3.19 Translating mechatronic device and friction observer

of the actuator force given to the system as an offset in parallel with its feedback controller so that the response performance can be improved almost 25%.

Another application of after-design simulation work in engineering systems is the pre-analysis and optimization of inputs embedded in computers of the manufacturing systems. Manufacturing of small parts can be costly and cumbersome since it often requires trial and error of adjustment of the machine settings. However, a remedy to this can be found by use of virtual iterative learning as reported in (Türeyen et al. 2016). A simulation of the additive-manufacturing system can be developed and used in parallel with a learning algorithm on the dimensional error of the final part before the real production is actually ran as shown in Fig. 3.20a. Researchers report using this method can improve the dimensional accuracy of a representative part up to 75% (Fig. 3.20b).

It is finally noted that similar to MDO based efforts for designing multidisciplinary systems such as automotive and aerospace products, there has been a continuous effort to design controlled mechanical systems using co-design strategies (Patil et al. 2010). The ultimate goal in these studies is to develop design frameworks that allow to reach system optimal designs from both the control and the mechanical design perspectives. Toward that goal, one such recent study is performed in (Kamadan 2016) where co-design strategies are proposed for robotic systems.

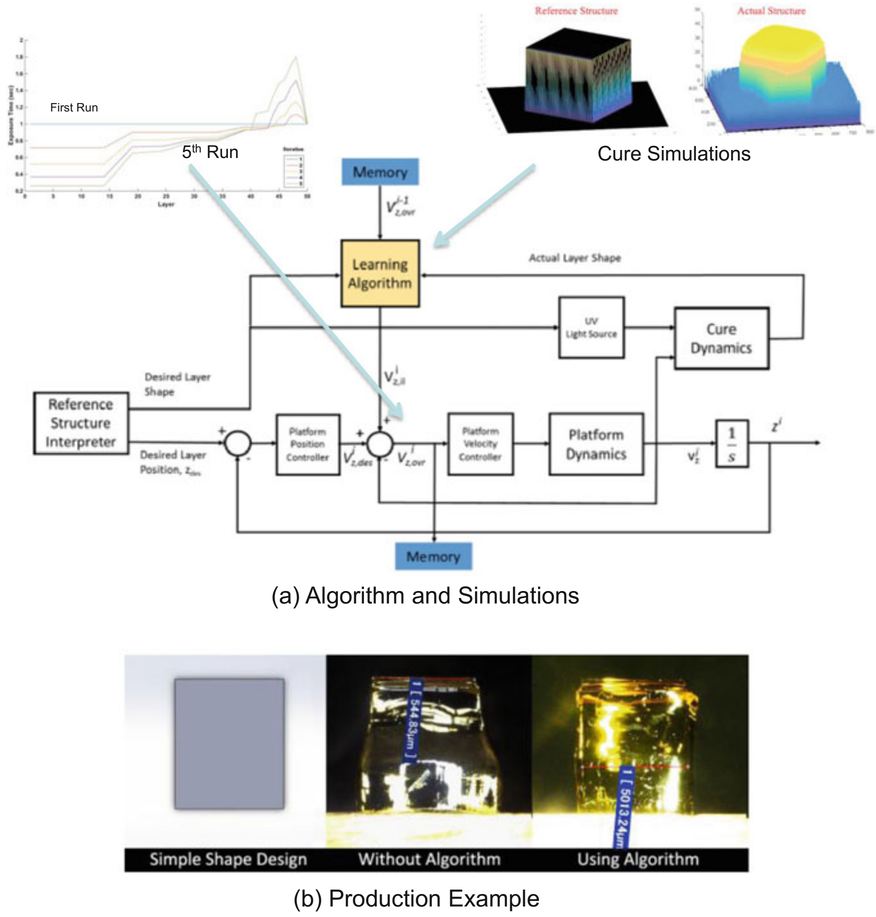


Fig. 3.20 Improving manufacturing quality with preproduction iterations

Robotic systems designed using domain-specific conventional approaches result in underperforming systems, i.e., are not system-optimal. This work introduces for the first time a unified framework of system-optimal designs of nonlinear controlled robotic systems driven by compliant actuators spanning a range of designs.

3.5 Conclusions and Outlook of the Topic

The ultimate objective within nearly all engineering projects is to reach a functional design without violating any of the performance, cost, time, and safety constraints while optimizing the design with respect to one of these metrics. Generally, in the beginning of each project, *wish list* like high-level requirements for the msyste performance are specified. Then, high-level requirements are cascaded down to the lower levels of the system allowing systematic design steps to be applied to these well-defined engineering design problems. The resulting problems are concrete problem constructs that contain quantifiable performance and constraint metrics. In time, two primary approaches emerged for the solution of complex engineering design projects. With the early approach also known as the "Waterfall Design Process", the subproblems can be tackled and solved sequentially. In recent years, as an extension of "Waterfall Design Process", a new approach has emerged called the "V-model" where scalable and varying fidelity simulations plays an important role before the actual prototype of the system can be build.

A good mathematical model is at the heart of each powerful engineering simulation being a key component in the design process. These models can be obtained by using physics-based methods, empirical collections and analysis or a combination of these two for balanced fidelity and complexity. Another important aspect of developing simulations is its resolution, or in other words its building blocks. In the simulation of the continuum, systems can be built from their smallest elements using the most fundamental forms of the governing equations. Sometimes, a lumped parameter-based simulation of machinery approach can be taken to simplify the simulations and the forthcoming engineering work such as in the case of the model based control system design.

The introduction of efficient and powerful platforms enabled researchers to solve/simulate the constitutive laws of continuum in mechanics in combination with the laws of conservation of mass, energy, and momentum. The same is valid for other fields including fluid mechanics and EM. Some of the most popular methods used for this purpose are the Finite Element Method (FEM), Finite Volume Methods (FVM), Finite-Difference Methods (FDM), and Boundary Element Methods (BEM). These methods are applied to the simulation of matter in all forms, i.e., solids, liquid, and gas, based on a major assumption of continuum media, thus Computational Mechanics of Continua. Namely, continuum describes the non-separability of the considered domain and validity of continuity between any points in the domain so that differentiation is possible. Therefore, continuity between elements in any continuum based numerical technique is maintained as well.

Many realistic problems present themselves as very complex problems due to their multi-physics and multi-scale nature. More specifically, scientists and engineers from various fields have been working on the combination of different numerical techniques with the goal of addressing these complex and elaborate physical processes, such as the transition from continuum to discontinuum (e.g., fracture processes) or the interaction of multi-phases of matter (e.g., hydrofracture processes) at micro–macro scales. As a result, new classes of numerical methods called hybrid or multi-physics and multi-scale methods evolved.

In today's world, developing multidisciplinary systems such as for instance cyber-physical systems that consist of both mechanical and electrical components constitute a key part of the engineering projects. These types of systems contain software algorithms, digital sampling and power electronics, mechanical components as well as communication networks that have to be developed concurrently to work coherently. This critical need for coherence has rapidly increased the importance of developing multi-domain simulations and engineers capable of supporting multidisciplinary analysis and design methodologies.

Review Questions

1. What are the primary phases of the engineering design cycle and how is simulation work used in each of them?
2. Name some of the numerical methods that exist to solve continuum problems.
3. Which type of problems can be classified as multi-physics and multi-scale continuum problems?
4. What challenges exist today when solving continuum problems?
5. What do all numerical methods suited to solve continuum problems have in common?
6. What are the effects of not-considering supporting hardware (networks, digital computers, etc.) to simulation performance in engineering systems?
7. What are possible uses for the simulations developed for the design cycle after product release to improve performance?

References

Ashby, M. F. (1996). Modelling of materials problems. *Journal of Computer-Aided Materials Design*, *3*(1–3), 95–99.

Automotive Suspension - MATLAB & Simulink Example. (n.d.). Retrieved February 15, 2017, from https://www.mathworks.com/help/simulink/examples/automotive-suspension.html?requestedDomain=www.mathworks.com.

Bathe, K.-J. (2006). *Finite element procedures*. Klaus-Jurgen Bathe.

Bensoussan, A., Lions, J.-L., & Papanicolaou, G. (1978). *Asymptotic analysis for periodic structures* (Vol. 5). North-Holland Publishing Company Amsterdam.

Cakmakci, M., & Ulsoy, A. G. (2009). Improving Component-Swapping Modularity Using Bidirectional Communication in Networked Control Systems. *IEEE/ASME Transactions on Mechatronics*, 14(3), 307–316. http://doi.org/10.1109/TMECH.2008.2011898

Cakmakci, M., Li, Y., & Liu, S. (2011). Model-in-the-loop Development for a Fuel Cell Vehicle. In *Proceedings of the American Control Conference* (pp. 2462–2467).

Chen, C.-T. (1995). *Linear system theory and design*. Oxford University Press, Inc.

Chew, W. C. (1995). *Waves and Fields in Inhomogeneous Media* (Vol. 522). New York: IEEE Press.

Chew, W. C., Michielssen, E., Song, J. M., & Jin, J.-M. (2001). *Fast and efficient algorithms in computational electromagnetics*. Artech House, Inc.

Dhatt, G., Lefrançois, E., & Touzot, G. (2012). *Finite element method*. John Wiley & Sons.

Dokuyucu, H. I. H. I., & Cakmakci, M. (2016). Concurrent Design of Energy Management and Vehicle Traction Supervisory Control Algorithms for Parallel Hybrid Electric Vehicles. *IEEE Transactions on Vehicular Technology*, *65*(2), 555–565. http://doi.org/10.1109/TVT.2015.2405347.

El-Kahlout, Y., & Kiziltas, G. (2011). Inverse synthesis of electromagnetic materials using homogenization based topology optimization. *Progress In Electromagnetics Research*, *115*, 343–380. doi:10.2528/PIER10081603 http://www.jpier.org/pier/pier.php?paper=10081603.

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2009). *Feedback Control of Dynamic Systems* (6th ed.). Prentice Hall. Retrieved from http://www.amazon.com/dp/0136019692.

Fung, Y.-C. (1965). *Foundations of solid mechanics*. Prentice Hall.

Gurel, L., & Ergul, O. (2007). Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns. *Electronics Letters*, *43*(9), 499–500.

Harrington, R. F. (2001). *Time-Harmonic Electromagnetic Fields*. *New York*. IEEE Press.

Jategaonkar, R. V, Fischenberg, D., & Gruenhagen, W. (2004). Aerodynamic modeling and system identification from flight data-recent applications at dlr. *Journal of Aircraft*, *41*(4), 681–691.

Kamadan, A. (2016). *Development of Co-design frameworks for optimal variable compliant actuation*. Sabanci University.

Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2000). *System Dynamics- Modeling and Simulation of Dynamic Systems* (Third Edit). Wiley-Interscience.

Kazachkov, I. V, & Kalion, V. A. (2002). Numerical Continuum Mechanics. Lecture notes. KTH.

Khurmi, R. S., & Gupta, J. K. (1976). *Theory of machines*. Eurasia.

Kiziltas, G., Psychoudakis, D., Volakis, J. L., & Kikuchi, N. (2003). Topology design optimization of dielectric substrates for bandwidth improvement of a patch antenna. *IEEE Transactions on Antennas and Propagation*, *51*(10), 2732–2743.

Lian, F., Moyne, J., & Tilbury, D. (2002). Network Design Consideration for Distributed Control Systems. *IEEE Transactions on Control Systems Technology*, *10*(2), 297–307.

Malvern, L. E. (1969). *Introduction to the Mechanics of a Continuous Medium.*

Martins, J. R. R. A., & Lambe, A. B. (2013). Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal*, *51*(9), 2049–2075. http://doi.org/10.2514/1.J051895.

Milton, G. W. (2002). The theory of composites (Cambridge monographs on applied and computational mathematics).

Muntean, A. (2015). *Continuum Modeling: An Approach Through Pratical Examples*. Springer.

Ogata, K. (1990). *Modern Control Engineering*. Prentice Hall.

Ogata, K. (1995). *Discrete-Time Control Systems* (2nd ed.). Prentice Hall. Retrieved from http://www.amazon.com/dp/0130342815.

Patil, R., Filipi, Z., & Fathy, H. (2010). Computationally Efficient Combined Design and Control Optimization using a Coupling Measure. *IFAC Proceedings Volumes*, *43*(18), 144–151. http://doi.org/10.3182/20100913-3-US-2015.00126.

Rajamani, R., Choi, S. B., Law, B. K., Hedrick, J. K., Prohaska, R., & Kretz, P. (2000). Design and Experimental Implementation of Longitudinal Control for a Platoon of Automated Vehicles. *Journal of Dynamic Systems, Measurement, and Control*, *122*(3), 470–476. http://doi.org/10.1115/1.1286682.

Rajamani, R., & Hedrick, J. K. (1995). Adaptive observers for active automotive suspensions: theory and experiment. *IEEE Transactions on Control Systems Technology*, *3*(1), 86–93. http://doi.org/10.1109/87.370713.

Ristevski, S., & Cakmakci, M. (2015). Mechanical design and position control of a modular mechatronic device (MechaCell). In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)* (Vol. 2015–August, pp. 725–730). IEEE. http://doi.org/10.1109/AIM.2015.7222623.

Schetz, J. A., & Fuhs, A. E. (2013). *Fundamentals of fluid mechanics*. John Wiley & Sons.

Sokolowski, J. A., & Banks, C. M. (2010). *Modeling and simulation fundamentals: theoretical underpinnings and practical domains*. John Wiley & Sons.

Topçu, O., Durak, U., Oğuztüzün, H., & Yilmaz, L. (2016). *Distributed Simulation*. Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-03050-0.

Tureyen, E. B., Karpat, Y., & Cakmakci, M. (2016). Development of an iterative learning controller for polymer based micro-stereolithography prototyping systems. In *2016 American Control Conference (ACC)* (pp. 852–857). IEEE. http://doi.org/10.1109/ACC.2016.7525020

Ulsoy, A. G., Peng, H., & Çakmakci, M. (2012). *Automotive Control Systems [Hardcover]*. Cambridge University Press. Retrieved from http://www.amazon.com/Automotive-Control-Systems-Galip-Ulsoy/dp/110701011X.

Volakis, J. L., Mumcu, G., Sertel, K., Chen, C.-C., Lee, M., Kramer, B., … Kiziltas, G. (2006). Antenna miniaturization using magnetic-photonic and degenerate band-edge crystals. *IEEE Antennas and Propagation Magazine*, *48*(5).

Walsh, G. C., Ye, H., & Bushnell, L. (2002). Stability Analysis of Networked Control Systems. *IEEE Transactions on Control Systems Technology*, *10*(3), 438–446.

Wang, G. G., & Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, *129*(4), 370–380.

Widder, D. V. (1976). *The heat equation* (Vol. 67). Academic Press.

World-Record Algorithm from Jülich Calculates Over Three Trillion Particles - Research in Germany. (2011). Retrieved February 15, 2017, from http://www.research-in-germany.org/en/research-landscape/news/news-archive/2011/08/2011-08-02-world-record-algorithm-from-j-lich-calculates-over-three-trillion-particles.html.

Zienkiewicz, O. C., & Taylor, R. L. (2005). *The finite element method for solid and structural mechanics*. Butterworth-heinemann.

Author Biographies

Melih Cakmakci is Assistant Professor of Mechanical Engineering at Bilkent University in Ankara, Turkey. He received his B.S. in Mechanical Engineering from M.E.T.U Ankara in 1997. He received his M.S. and Ph.D. in Mechanical Engineering from University of Michigan in 1999 and 2009 respectively. His research areas include modeling, analysis and control of dynamic systems. Prior to joining Bilkent University, he was a Senior Engineer at the Ford Scientific Research Center at Dearborn, Michigan, USA. He is a member of ASME, IEEE, and SAE.

Gullu Kiziltas Sendur is Associate Professor with the Mechatronics Engineering Program, Sabanci University. She received the B.Sc. and M.Sc. in Mechanical Engineering from the Middle East Technical University, Ankara, in 1995 and 1998, respectively, and the Ph.D. in mechanical engineering from the University of Michigan, Ann Arbor, in 2003. She was a Postdoctoral Researcher with both the Electro-Science Laboratory, Ohio State University and the University of Michigan. She also coordinated with the Ceramic Research Group, Material Science and Engineering Department, University of Michigan, on the advanced fabrication of dielectric composites. Her research efforts center on the design, analysis, and fabrication of complex engineering systems, such as miniaturized electromagnetic, electromechanical, and biomedical devices and multidisciplinary design optimization techniques. She has published a book chapter, numerous journal and conference articles and was honored with Career Award by TUBITAK (2007) and Young Scientist Award by the Science Academy of Turkey (2008). She is a member of the IEEE, ASME, and ACerS.

Umut Durak is a research scientist in the Institute of Flight Systems at the German Aerospace Center. He is also Adjunct Lecturer in the Department of Informatics at Clausthal University of Technology. His research interests include model-based simulation engineering, simulation-based systems engineering and ontologies in simulation. He received his BS, MS and Ph.D. degrees in Mechanical Engineering from Middle East Technical University (METU) in Turkey. He is a member of the Society for Computer Simulation International (SCS) and Arbeitsgemeinschaft Simulation (ASIM) and a senior member of the American Institute of Aeronautics and Astronautics (AIAA). And, he is a part of AIAA Modeling and Simulation Technical Committee and editorial teams of the Simulation: Transactions of the Society for Modeling and Simulation International; and the International Journal of Modeling, Simulation, and Scientific Computing.

Chapter 4
Simulation-Based Systems Engineering

Andreas Tolk, Christopher G. Glazner and Robert Pitsko

Abstract Systems engineering (SE) is understood as an interdisciplinary collaborative approach to derive, evolve, and verify a life cycle balanced system solution. Satisfying customer expectations has become more complicated and complex, as we are no longer designing systems optimized for a single point, but instead are focusing on systems which are sustainable, resilient, flexible, and even antifragile. Simulation is established to support analysis and testing of systems. Recent developments—the use of executable architecture concepts allowing for dynamic evaluation of system concepts, and the use of agent-based implementation to support learning and adaptive system behavior—allow better support of an agile enterprise. To support these ideas, simulation must be fully integrated into SE paradigm. We must establish simulation as an integrated discipline within the SE methodology. Research is needed to support validation and verification of self-modifying systems, as well as improved heuristics for computationally complex problems. This chapter proposes detailed visions and identifies a research agenda needed to realize these visions accordingly.

Keywords Agent-based paradigm · Antifragile · Complex adaptive system · Executable architecture · Foundational subset for executable UML models (fUML) · Functional mockup interface (FMI) · High fidelity modeling · -ilities · Machine learning · Meta-model · Tradespace · Validation

---

The authors affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved for public release: Case: 17-1250.

A. Tolk (✉)
The MITRE Corporation, Hampton, VA, USA
e-mail: atolk@mitre.org

C.G. Glazner · R. Pitsko
The MITRE Corporation, McLean, VA, USA
e-mail: cglazner@mitre.org

R. Pitsko
e-mail: rpitsko@mitre.org

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_4

4.1 Introduction

Many professional organizations, like the Institute of Electrical and Electronics Engineers (IEEE) and the International Council on Systems Engineering (INCOSE), provide slightly different definitions, but generally systems engineering is understood as an interdisciplinary collaborative approach to derive, evolve, and verify a life-cycle balanced system solution. Satisfying customer expectations has become more complicated and complex, as we are no longer designing systems optimized for a single point, but instead are focusing on systems which are sustainable, resilient, flexible, and even antifragile. These, and other characteristics, are collectively known as the '-ilities.' Increasing use of Model-based Systems Engineering (MBSE), as introduced by Wymore (1993), to replace documents with a common model with many facets has improved aspects of the systems engineering challenge. However, it continues to require a broader understanding of the system of interest in the context of the larger environment. We need to understand the behavior of a system in its socio-technical context, extend the ideas of Booher (2003) and bring them to the next level. The core of this chapter is to highlight how simulation can support these tasks!

Simulation has long been used as a key tool in domain specific system design and analysis. The paper of Smith (1962) is an early example. Wymore's (1967) ideas influenced the development of the modeling and simulation specification languages, such as GEST (Ören 1984b), which was, since its beginnings in 1969, based on system theoretic concepts. The often used Discrete Event Simulation (DEVS) formalism is based in systems engineering principles (Zeigler et al. 2000). Only recently, compendia on simulation-based systems engineering were published (Gianni et al. 2014; Rainey and Tolk 2015). Without doubt, simulation contributes to successful many systems engineering solutions.

It is therefore surprising that simulation has not made its way into the toolbox of systems engineers and into the context of model-based systems engineering as a general solution method, but merely as a set of tools. While engineers do use simulation methods to gain numerical insight into the dynamic behavior of a system, developing and applying this understanding in the context of engineering the system is generally not supported by systems engineering methods. The multitude of simulation tools is not yet accompanied by a coherent systems engineering simulation method that guides their application, such as recommended recently by D'Ambrogio and Durak (2016). If considering the dynamics of a system occurs early and throughout the engineering process, undesired behaviors can be identified and addressed earlier in the life cycle, where changes can be made at favorable cost points. If this is done following a common method will ensure discovery of shortcomings and identification of alternative solutions timely. By providing examples of the state of the art and making observations on what can be accomplished in the near future, the authors hope to contribute to closing this gap.

After a discussion of the prevailing systems engineering practice and the current incorporation of model-based techniques into the practice, this chapter highlights system engineering challenges that require something more than a rigorous, repeatable, verifiable modeling scaffolding. There is a need for a dynamic understanding of behaviors and interdependencies within its broader context. Similarly, simulation is established to support analysis and testing of systems, but has often been limited to within a single engineering domain or has narrowly defined and modeled system performance, implemented as tool specific add-ons. Recent developments—in particular the use of executable architecture concepts allowing for dynamic evaluation of system concepts and the use of complex adaptive modeling approaches to support learning and adaptive system behavior—allow better understanding of system behavior in a much broader context. Related examples, motivations, and definitions have been introduced by Pawlowski et al. (2004), Wheeler and Brooks (2007), and Tolk and Hughes (2014).

Related to this topic is the domain of modeling and simulation-based systems engineering (Gianni et al. 2014) that compiled applications of the simulation-based discipline on a broader sense, generally showing how model-based approaches can improve current methods, including all flavors of models, such as business process modeling, architecture description language, and more. Coping with all these ideas in this chapter, however, would go far beyond our scope, so that focus on examples is needed.

To elevate simulation-based systems engineering to the next level, the authors focus on two topics within this chapter.

- First, the use of situated executable architectures—the execution of a virtual system directly derived from its systems engineering specification in its operational context—will not only allow for an element of empirical insight into the dynamic behavior of the system in various environments, but will also provide system engineers and potential users a fully immersed experience of the new system, helping to identify unstated user preferences and requirements earlier in the engineering process.
- Second, using the research results of agent-based modeling and human-in-the loop simulation, system engineers will be able to explore and understand how their systems will behave when given new environmental and operational constraints, and apply self-organizing and learning algorithms accordingly.

To support these ideas, simulation must be fully integrated into the systems engineering paradigm. This will require the evolution of MBSE, which is largely focused on modeling, to rely on simulation. In other words, not only do we need standards that allow for better collaboration between MBSE and simulation, we must also establish simulation as an integrated discipline within the systems engineering methodology. Furthermore, research is needed to support validation and verification of self-modifying systems, as well as improved heuristics for computationally complex problems. This chapter proposes detailed visions and identifies a research agenda needed to realize these visions accordingly.

4.2 State of the Art in Model-Based Support to Systems Engineering

> **Optimistically**, the maturing of system engineering into a recognized discipline from its roots in large aerospace and defense programs has been, and will remain, an enabling factor in the ability of societies to deal with the macroscale problems facing us in energy, environment, and other key areas. **Pessimistically**, system engineers have some explaining to do. How is it that we continue to encounter failure of important and complex systems where everything thought to be necessary in the way of process control was done, and yet despite these efforts the system failed? Each time this occurs, we as an engineering community vow to redouble our efforts to control the engineering process, and yet such events continue to occur. The answer cannot lie in continuing to do more of the same thing while expecting a different outcome.
>
> Mike Griffin, Recent NASA Administrator. (Griffin 2010)

4.2.1 Integrating Modeling into the Systems Engineering Methodology

Systems engineering seeks to respond to some 'market opportunity' or 'need' with a solution. There is a sense of novelty, in that something new can be envisioned and built in part because of advances in technology, knowledge and enabling techniques. Often this requires trial and error or experimentation with existing technology being applied in new ways.

At the turn of the twentieth century, the Wright brothers were searching to understand the exact action of the propeller. They were pursuing a system for human controlled flight and intended to combine an engine with a propeller for propulsion. Understanding the principles and the subsequent design ramifications of making and spinning a blade in such a manner so as to propel an airplane required engineering ingenuity. The Wright brothers spent significant time wind tunnel testing physical models to understand the performance curves of the propeller as well as significant amounts of physical 'hands on' trial and error to understand the engineering constraints of integrating an engine, propeller and airplane system together (McCullough 2015).

The Wright brothers were ultimately successful, inspiring the world, and together with other aviation pioneers together they catalyzed an industry. The Wright brothers' system success provides three themes we explore in this section: (1) scale of system specification, (2) performance expectation, and (3) trial and error as a vital part of system success. For the 1903 Wright Flyer, these themes were modest: the system specification was able to be completely known by two people, the performance expectation was one of system possibility in a static environment, and trial and error required could be safely and efficiently conducted by advancing physical scales during a campaign of experimentation and demonstration.

However, as the twentieth century progressed in technology, knowledge, and enabling techniques the corresponding ‘market opportunity’ or ‘need’ required advancement in Systems Engineering. No longer could the system specification be completely known by a few people. Such specification had to be expressed via shareable artifacts. The performance expectation of these systems demanded increasing precision as well as envelope pushing results in both a knowable and unknowable environment. The trial and error expected in these system contexts exceeded what could be expected by a campaign of physical experimentation and demonstration alone.

4.2.1.1 Systems Engineering of High Performance Systems

At the midpoint of the twentieth century, the world became captivated with space. A space race among world powers created a ‘market opportunity’ or a ‘need’ for human spaceflight and exploration. Rapidly advancing technology, knowledge and enabling techniques required a unifying systems engineering approach to pull it all together. Brill (1998) points to the NASA Apollo program as among the first successful applications of such systems engineering.

The Apollo program was one of a series of programs designed to deliver safe human spaceflight (Kranz 2001). The scale of these programs required NASA to create procedural regulations to coordinate the efforts of multiple government, industry, and academic teams working both technical and operational aspects of the program. This coordination was facilitated by numerous technical requirement documents and checklists and culminated ultimately in what have become known as NASA Procedural Regulations. Several independent industry partners would be required to deliver subsystems for testing and ultimately final assembly leading to the launch and successful mission accomplishment of one of the most astounding and high performing systems in the world at that time.

On the way to achieving the performance expectation for human spaceflight, NASA integrated multiple scientific and engineering disciplines (e.g., materials science, aerodynamics, mechanical engineering and computer engineering…). Each of these disciplines relied on independent high fidelity modeling using physical and synthetic representations of critical aspects of a given subsystem of interest with corresponding environmental conditions. These models increased understanding of a specific property under examination, while generally in isolation of the larger system context. Frequently, computational extension of physical and theoretical results provided insights into expected subsystem performance; however, the aggregation of this high fidelity point analysis in the context of the larger system and environment was left to the senior engineers as a human activity.

Ultimately, NASA successfully delivered astronauts to the moon and back with the Apollo missions. This scientific and engineering achievement was enabled by nine years of development, testing, and a campaign of preparatory missions. These missions incrementally tested technology, procedures and the environmental extremes of space. Although this success represented the pinnacle in systems engineering at that time, the increasing role of software in systems and advances in computing technology set conditions for improvements.

4.2.1.2 Improving System Engineering with Model-Based Systems Engineering

While at NASA's Jet Propulsion Laboratory, Jeff Estefan provided a survey of Model-Based Systems Engineering (Estefan 2007). In this seminal paper on the topic, Estefan described Model-Based Engineering as "elevating models in the engineering process to a central and governing role in the specification, design, integration, validation, and operation of a system."

Due in part to increased formalisms in architecture and design as embodied in Unified Markup Language (UML) and Systems Markup Language (SysML) as well as the emergence of supporting tooling, modeling has improved Systems Engineering.

Instead of relying on static requirement documents and independent models of select aspects of subsystem performance as has been done previously. Model-Based Systems Engineering integrates and automates requirements, architecture, performance, and testing procedures. This can improve communication across distributed teams (Haskins 2011) as the system of interest becomes more complicated and requirements or performance objectives change. High fidelity modeling, tradespace analysis and physics-based performance simulation can inform detailed component design and corresponding operational enhancements leading to exceedingly high performing point solutions. In addition, Model-Based Systems Engineering can incorporate these high fidelity performance results into a larger system context where multiple stakeholders such as developers, engineers, operators, and managers can interact with the same model according to their perspective (Wheeler and Brooks 2007; van Cleave 2010).

4.2.1.3 Improving Model-Based Systems Engineering with Executable Architectures

Leading edge applications of Model-Based Systems Engineering, frequently referred to as 'executable architecture,' include a simulated and integrated system environment directly connected to the Model-Based Systems Engineering design environment (Wheeler and Brooks 2007). This results in a realistic, albeit synthetic, system environmental context to explore the performance and operational impacts of various subsystem design alternatives within a mission thread. Several approaches, as described by Wagenhals et al. (2009), often used formal methods, such as Petri Nets, to support these ideas. Other pioneers started to apply simulation formalism, as they will be discussed in the next section of this chapter, to support systems engineering artifacts (Mittal 2006). Allowing for near real-time system level operational performance evaluation based on results from various subsystem high fidelity 'physics-based' simulations improves understanding of anticipated system behavior in the design phase of the project, as opposed to waiting for traditional physical testing or integration phases of a project (Campbell et al. 2015).

Executable architectures were used during the redesign of surveillance and communication systems for the Air Force E-8C Joint Surveillance Target Attack Radar System (known as Joint STARS) aircraft. By converting static pictures of system and component architectures "into a programming language that can be compiled like a regular software program, such an executable model shows us how the parts interact during the design phase rather than waiting for software to be built." Using Model-Based Systems Engineering and executable architectures allowed the team to rapidly "blend multiple disciplines in a project and collaborate with multiple organizations." This resulted in a system-wide context understandable by systems engineers, developers and 'non-systems engineers,' alike allowing for the execution of mission threads and the evaluation of design options in near real time. This advancement allowed "the Air Force to determine how and where it wants to have an operator in the loop and where it feels comfortable letting the computers make decisions" (van Cleave 2010).

Executable architectures rely on "creating developer-level interface definitions and encoding business logic to enable component-level integration" (Campbell et al. 2015) within the larger system modeling environment. In order to faithfully represent the intended design, the Model-Based Systems Engineering specifies Application Program Interfaces, detailed message formats, and schemas for component level or subsystem integration. This allows the systems engineering teams to rapidly evaluate proposed design alternatives for performance behavior as well as verification and validation in the context of the larger integrated system. "Generating software from [such a] model of the interfaces helps capture the details of how the resulting implementation will interact with the larger system. This code can be used in the final product and will assure consistency with design." (Campbell et al. 2015) This advancement enables virtual 'wind tunnel' testing, where engineers can explore design options at the subsystem or component level in the modeling environment way before any system is physically constructed. Additionally, with Model-Based Systems Engineering with executable architectures engineering teams can now evaluate expected large scale system behaviors based on the aggregation of high fidelity 'physics based' models structured around expected mission scenarios.

4.2.2 The Limits to Modeling in Systems Engineering

Modeling provides dramatic insights into expected system performance. From the physical propeller models used by the Wright brothers, to the mathematical models of the NASA spaceflight programs, to the integrated and executable architectures of the E-8C Joint STARS Aircraft, modeling has enabled the design and development of high performing systems. Integrating modeling in the Systems Engineering process enables a deeper understanding of a given design element (or set of elements) in relation to the operational environment.

Consider advanced air superiority systems such as the F-22 (Holder and Wallace 1998). While these systems push the state of the art in fields such as aerodynamics, propulsion, materials, computing, the result remains a high performing, albeit, singular point system. The system context is largely static in that the constraints and expectations regarding flying in earth's atmosphere are well known and deterministic. Additionally, the operational context and expectations for both the pilot and the system are constant at the time of design. This aircraft was designed for multi-role air superiority against a field of near peer adversaries. Including modeling in the System Engineering process for this class of systems continue to enable efficient, effective and spectacularly high performing systems. This example of a particular defense system illuminates another general challenge: systems engineers create systems whose development cycles are so long that by the time they are fielded, the system's environment has fundamentally changed. An example is the political changes in the defense domain after the end of the Cold War. Nonetheless, systems exquisitely designed for operations in the Cold War era still had been introduced, and now are used in a context for which they were not designed, likely leading to subpar performance in a new, often not foreseeable class of new operations.

There is another class of systems where increasing the fidelity and integration of modeling within Systems Engineering is not enough to guarantee operational objectives. The design of these complex systems, as pointed out by NASA's Administrator Griffin, do not benefit from increased engineering process. This class of systems generally has a broader systems engineering context where system performance and environmental constraints cannot be fully predicted or understood. Often these systems

- Do not benefit from a point optimal solution or configuration,
- Require contribution from many sources (system and human) to achieve operational effect,
- Can benefit or suffer from second- or third-order effects stimulated from outside of their span of control, and
- Are expected to deliver value in environments which are not deterministic or are evolving faster than the system can be fielded.

Just as integrating high fidelity physics-based models can enable the systems engineering of the first class of high performing point systems, integrating simulation models into the systems engineering process can also enable the design of this second class of complex, open, and socio-technical systems.

4.3 Simulation in Support of Systems Engineering

> **Simulation**, if viewed from the right perspective, can be a discipline central to many others. **Knowledge** about existing and conceived **systems** represented with their static and dynamic structures and expressed in **computer processable forms** leads to comprehensive knowledge processing abilities many of which are germane to computerized simulation.
>
> Tuncer Ören, Simulation Pioneer (Ören 1984a)

While advances in model-based systems engineering have expanded the system engineering toolbox, such as allowing systems engineers to link to simulation analyses, we must promote system simulation to an equal footing with system modeling if we hope to be able to engineer the complex systems of the future. The gaps in our ability to engineer complex, adaptive systems that we have highlighted cannot be closed via the application of our current tools. We must expand our toolset to allow us to analyze the dynamic behavior of our adaptive systems in the uncertain, evolving environments of the future that we are designing them for. The simulation-based systems engineering approach addresses these shortcomings by augmenting our ability to perform high fidelity simulation analysis in a given context with a spectrum of performance analysis based on a plurality of simulated states and interactions. To engineer for the -ilities (Ross et al. 2008), we must develop simulation tools that allow for architectural analysis in environments where both the system and the environment are adapting.

4.3.1 Integrating Simulation into the Systems Engineering Methodology

As we have seen, simulation has long been used to help engineers of all disciplines understand how a component or system may perform in any given environment or scenario. Simulation, when used as part of a Design of Experiments, is frequently used to identify ideal, 'point solution' designs, explore the design space, and build an understanding of tradeoffs among design parameters. Such exploration can identify robust designs, such that if the system is not performing to specification or if the environment is slightly different than anticipated, system performance remains acceptable. This approach, however, does not assume complex, adaptive systems or environments, as many modern systems are. As opposed to an aircraft that is designed with performance analysis of its flight dynamics, a swarm of micro-UAVs should be able to respond collectively to unknown, hostile environments that will try to disrupt the swarm. This is a very different challenge that requires an updated systems engineering approach. Given the case of a single aircraft, we may wish to design the aircraft to be highly resilient to an evolving cyberattack, which may involve cyber elements or attack elements incompletely understood when the aircraft was specified or designed. To build an understanding of complex, adaptive systems' responses to such environments, we need to integrate simulation more effectively into the system engineering methodology.

Simulation-based system engineering addresses the shortcoming of the traditional systems engineering approach by augmenting point performance analysis with a spectrum of performance analysis based on a plurality of simulated states and interactions in situated and immersive environments. This approach provides the analytic and probabilistic framework necessary to explore implications of engineered complex adaptive systems where constituents may adapt their behavior and interact with the system in unanticipated ways. The objective of this approach is to extend our current model-based systems engineering analytical methods with richer analyses of the system's dynamics and its interactions with the larger, changing environment. For today's learning and self-adaptive systems, simulation based systems engineering provides engineers with a spectrum of analytical capabilities that are not available with classic or model-based system engineering methods. To achieve this vision, two critical components are required. First, we must begin to build better simulations of systems that allow us to understand their dynamics behavior. The second is that we need to improve how we evaluate those simulations in the context of their environments, through the use of both situated and immersive environments.

4.3.1.1 Insight into the Dynamic Behavior of Systems

Even for relatively straightforward and deterministic systems, humans have poor intuition regarding dynamic system behavior. A model that makes perfect sense when presented in its static views may reveal significant shortcomings when being executed or—in the context of this chapter—being simulated. For even simple feedback processes, people are poor at predicting system performance. Wagenaar and Timmers (1978, 1979) demonstrated that individuals poorly predict the behavior of systems with exponential growth. Even when presented with additional data points and graphs, people tend to linearly extrapolate. For slightly more complex systems, both Sterman (1989) and Deihl and Sterman (1995) experimentally demonstrated that people given complete information about relatively simple systems with feedback were poor at predicting system behavior and slow to learn. Simulation helps us work through these cognitive roadblocks by logically evaluating executable models of the systems that describe structure and behavior. As systems become larger, more driven by feedback, and less deterministic, the need for simulation to gain insight into the dynamic behavior of systems only increases. To support a vision of simulation-based systems engineering, we must continue to advance our ability to simulate systems in ways that help us understand the causal nature of their dynamics through the engineering process.

Our use of computer-based simulation to help us understand engineering systems goes back to the birth of the digital era, and has continued to advance since that time. We have seen the application of simulation as far back as the 1950s to what should be stable, intuitive systems, such as supply chains and industrial dynamics. Forrester (1961) demonstrated that even these 'stable' systems are subject to what has become known as the 'Bullwhip effect,' where delays and information asymmetry give rise to highly unstable dynamics. We have seen the proliferation of physics based simulations, the simulation of nonlinear dynamical systems, discrete event simulations, and more recently agent-based simulations all advance in their ability to simulate a system's behavior over time. Increased fidelity of simulation, together with improving experimental designs to evaluate a trade-space, help engineers identify critical performance parameters, stable and unstable regimes of behavior, the associated tipping points, and most importantly, it helps them refine their understanding of the aspects of a system's design that can have the most impact on the system's behavior.

The context in which we are designing systems is changing, however. Increasingly, systems are being asked to perform in missions for which they were not originally intended to serve, or must be designed to intelligently adapt in the face of a changing environment. The long lead time to design, build, and acquire complex systems has meant that a fighter aircraft or a sensor platform may be asked to serve in environments and in roles that look very different than the ones engineers originally planned for. While we may have developed very useful simulations of the flight performance characteristics of that aircraft, our current systems engineering processes are not adequate for us to use those simulations to understand how that aircraft can best be designed to most effectively adapt to changing requirements. For this, we need to advance how we evaluate simulations of our systems in more realistic, situated, and immersive environments. This is going to allow to evaluate the usefulness of the provided system model using simulation methods as an integrated tool of the systems engineering process.

4.3.1.2 Using Situated Simulation Environments

Situated environments are those that allow a system simulation to be placed in its operational context to observe of the system responds to that environment. The intent of a situated simulation environment is to create a realistic representation of a system and its environment in a given operational context. For example, a simulation of an off-road vehicle may be placed in a simulated desert environment, or later placed in a temperate rainforest. By placing a simulation in a situated environment, engineers can learn more about potential system performance and behavior than can be understood by looking at simulation output parameters such as torque, acceleration, and mass devoid of their context. We can move to asking higher level questions, such as "Can the system execute its mission?" or "Is system A more adaptable than system B in a certain context?" or "Which design is most robust to a desert environment?" or "How did the system perceive and adapt to the environment?"

Another important aspect is the possibility to evaluate the operational effectiveness of a system in the context of the portfolio it is operating in against a set of potential opponents under different environmental conditions. Garcia and Tolk

(2015) extend the work of Mittal (2006) by incorporating context from of a situated operational environment into the executable artifact describing the system behavior. In their example, they describe a system that fully fulfilled its requirements and delivered good performance, but in the context of its portfolio, used too many resources that were no longer accessible to its partner system. This resulted in an overall decrease of the operational effectiveness of the portfolio in which the system was supposed to operate. Many of these insights are obvious in the aftermath, but nonetheless easy overlooked without the support of an operational context.

Such integral use of a simulation environment as part of the systems engineering process should increase the likelihood that the systems developed will meet users stated and unstated preferences when fielded and will easily meet those preferences as future environmental changes occur, unlike many of our systems today. As we build ever more adaptive and autonomous systems, we must move to using situated environments to evaluate their rulesets and situation adequate behavior, including observability of potential emerging behavior (Tolk 2015).

4.3.1.3 Immersive Environments

Immersive simulation environments differ from situated environments in that users are brought into the analysis loop and directly experience the simulated system. This can be done in a serious gaming context, or using rapid emerging tools from virtual and augmented reality. This allows users to develop a deeper, intuitive understanding of a system's dynamic behavior in response to differing environments, and more directly allows them to uncover unstated preferences and system requirements. Some of these technologies exist in a training context, such as flight simulators. They are often very high fidelity, but they are used to train pilots, rather than upstream in the design process to design the aircraft itself and improve its potential performance before the first aircraft is prototyped.

In an immersive environment, the focus of evaluation is on the experience of the system in its context. We have seen that people learn best through experiential learning. For large, complex engineered systems, this can be very expensive or potentially dangerous. A test pilot is an inherently risky profession, and operational testing is very expensive. Simulation with immersive environments very early in the systems engineering life cycle allows the system to be changed while it is still relatively inexpensive to do so. Once physical prototypes of large complex systems are built and evaluated, it is often prohibitively expensive to make substantive design changes. A virtual test pilot can crash thousands of virtual aircraft, learning from each virtual aircraft's flaws, far more quickly, safely, and at dramatically less cost.

The key to immersive environments is experiential learning. How can systems engineers and future users build a deeper understanding of a complex system's emergent behaviors and dependencies in different environments? This must be done earlier in the system engineering process to have the greatest impact, and ideally should leverage existing model-based systems engineering artifacts. This requires advanced, human-in-the-loop simulation capabilities. These may involve highly visual approaches, or may derive the immersive nature from the feedback, such as in a 'Management Flight Simulator' (Sterman 1992). Management flight simulators have been developed to help discover, understand, and manage organizations, but not yet as an input to designing them. More visually immersive environments are likely to come from advances in the video gaming community. For example, the video game 'America's Army,' intended as a recruitment tool, has enough fidelity in its physics engine to serve as a faithfully situated simulation environment for a number of defense systems (Zyda 2005). This allows games to be far more than games. Other experimental capabilities, such as MITRE's Center for Advanced Aviation System Development's 'Idea Lab,' allow human-in-the-loop simulations of air traffic control, aircraft, and airport operations to be quickly combined in different situated contexts to evaluate new concepts of operation for the Federal Aviation Administration.

The key to capturing life cycle savings and promoting our ability to explore far more designs is by incorporating the use of situated and immersive environments with system simulations earlier in the system engineering process. The technology exists, or is rapidly evolving; we need to leverage them for system engineering, in addition to their current use for training and even entertainment. This approach will allow us to do a better job identifying designs and architectures which meet our stated and unstated requirements earlier in the systems engineering processes, increasing our satisfaction with our systems at reduced costs.

4.3.2 Applying Complex Adaptive System Technology to Improve Systems Engineering

While the first subsection described the benefits and a way forward for a full integration of simulation to the Systems Engineering methodology, this subsection will describe how the application of complex adaptive system technology within system design will improve the results of systems engineering by enabling resilient, self-healing, and even self-improving systems. At the core of these ideas is the use of methods as provided to implement capabilities of intelligent software agents, as defined by Tolk and Uhrmacher (2009), to replace functions of the system. By doing so, the system will be able to expose the characteristics of complex, adaptive systems, including learning, adapting, and improving its own capabilities.

4.3.2.1 Agent-Based Metaphor: Self-organization, Learning, and Social Behavior

In an extensive literature review, Tolk and Uhrmacher (2009) compiled the various definitions of intelligent software agents, as they are used for agent-based modeling approaches. The emergent common understanding has been that of an intelligent, independent software object, which adds decision layers between the request of function and its execution: when objects receive the request to execute one of their functions, they do so; when agents receive the request, they may decide to provide the functions, or an alternative, or not react at all. That depends on how the agent perceives its environment, and what internal objective an agent follows. Jennings et al. (1998) characterized this behavior illustratively as "*objects do it for free, agents do it for money.*" Overall, the following characteristics were recognized in a multitude of evaluated proposed definitions:

- ***The agent is situated, it perceives its environment, and it acts in its environment***: The environment includes typically other agents, other partly dynamic objects, and passive ones, that are, e.g., subject of manipulation by the agent. The communication with other agents is of particular interest systems comprising multiple agents, as agents can collaborate and compete for tasks. This latter characteristic has also been referred to as social ability.
- ***The agent is autonomous***: An agent can operate without the direct intervention of humans or others. This requires control about its own state and behavior. Agents are guided by some kind of value system, which is expressed in form of computable utility functions that are used as metrics to evaluate options.
- ***The agent is flexible and can learn***: To be flexible for an agent means to mediate between reactive behavior, being able to react to changes in its environment, and deliberativeness to pursue its goals. A suitable mediation is one of the critical aspects for an agent to achieve its tasks in a dynamic environment. An agent can act upon its knowledge, its rules, beliefs, operators, goals, and experiences, etc., and adapt to new constraints and requirements —or even new environments—as required. For example, new situations might ask for new goals, and new experiences might lead to new behavior rules.

Figure 4.1 demonstrates these agent characteristics.

The agent has a value system describing his goals, beliefs, and desires. This value system drives all decisions. The decisions are based on the perception the agent has of its situated environment. This perception may be incomplete and even wrong, depending on its sensors. To improve the perception, but also for planning purposes, the agent can communicate with other instances before he acts on elements of the environment. Finally, an agent may use simulation to support its perception as well, to add the dynamic evaluation components described earlier in this chapter.

The principles of machine learning based on model-based principles was described by Zeigler (1986). It describes how a perception is created, and how this perception is interpreted.

- An agent has sensors that observe its environment. These sensors do not observe everything, but have physical and technical constraints (actually, they are systems themselves). These sensors are purposefully selected by the System designer to collect relevant data needed to inform potential actions.

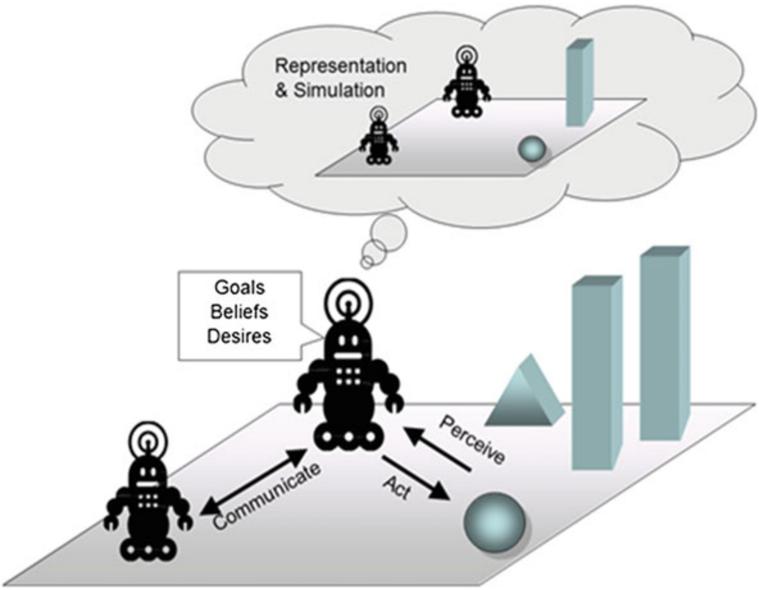


Fig. 4.1 Agent characteristics

- The observation of the sensors is mapped and fused into a perception. This perception is limited by what the sensors can observe, the quality of the fusion algorithm, and the overall perception model: if there is no place in the perception model to map an observation to, the observation data cannot become part of the perception model.
- The awareness of the system of its environment becomes the intersection of sensor and perception models, additionally constrained by the quality of the fusion and mapping algorithms. Gaps may be closed by communication with other agents, who may have observed additional information.
- A system can only recognize what options are available in a set of meta-models that combine observable situation and required action. These can be parameterized to capture a group of functionality related situations, e.g., combining speed of a car and required braking force to stop in time before hitting an obstacle. These can actually be executable meta-models, or simulations of the recognizable object.
- This perception must be mapped to a set of known situations the system can take action (this can be called the action model), capturing the observable environment status to a set of action parameters. If the observed state does not match any of those possible situations, the default action is chosen (often: do nothing). Also, a fuzzy interpolation between most similar surrounding situations is possible, or an extrapolation from the most similar neighbored situations.

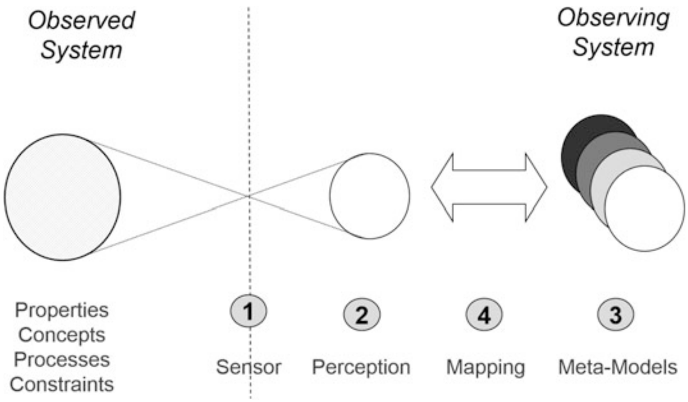


Fig. 4.2 Machine understanding

Figure 4.2 depicts these principles of machine understanding.

It is worth mentioning explicitly that the meta-models follow the principles of simulations. Each meta-model is a representation of a recognizable object. If something is recognized as such, the methods assigned to this object are executable to provide the observed objects with the dynamics of the recognized object. If discrepancies are observed between the observed behavior and the recognized behavior, the mapping may need to be reevaluated, i.e., if something looks like a known object from the observed attributes but does not behave as expected, it may either be something different, or the possible behavior patterns may have to be expanded.

In this context, learning means (1) to find better calibration parameters for existing meta-models, (2) to create new meta-models if new concepts are needed, or (3) to expand the meta-models to include new insights. It also means to perpetually improve the utility functions that represent the value system.

While traditional systems engineering defines systems functionality, incorporating the ideas described above allows systems to observe, communicate, and learn. This allows them to improve their function, recover from attacks, and even learn to recognize new concepts and apply them for sense and decision making.

4.3.2.2 From Stability via Sustainability to Antifragility

The objective of systems engineering is to provide a quality product that responds to a market opportunity or meets the user needs. However, these user needs have evolved over time, leading to a collection of 'ilities' that have to be exposed as characteristics by the system, often used to describe the nonfunctional or system-wide requirements for a system. In the context of this chapter, the focus lies on stability, sustainability, and antifragility.

- **Stability**, also referred to as *robustness*, is understood as being able to continually deliver expected performance across a set of all expected—and in particular negative—environmental conditions. This may also require some degree of *flexibility* allowing the system not to break under slightly different conditions, still being able to deliver appropriate performance.
- **Sustainability** is the ability to deliver expected performance over a long period of time, even under extreme circumstances. Often, this implies *resiliency*, which is the ability to recover rapidly from negative influences that may result in a temporary incapacity of the system, like an attack or a catastrophic event.
- **Antifragility** is a property of systems that increase in capability, resilience, or robustness as a result of stressors, shocks, volatility, noise, mistakes, faults, attacks, or failures, or, as Taleb (2012) defines them: systems that get better under stress.

Other related 'ilities' often required in this contexts are adaptability—being able to change based on foreseen internal stimuli—and agility—being able to change within a relevant time horizon. De Weck et al. (2012) give an overview of the most common used 'ilities' and their semantic relations.

Within this chapter we postulate that simulation-based systems engineering can help move systems from being stable/robust to become sustainable/resilient and ultimately antifragile.

It is well known and accepted that simulation provides an enduring optimization capability, where the system of interest, the environment and potential future states are explored and evaluated. When the environment changes (e.g., due to an expected or unexpected stressor) the system of interest is positioned within the simulation to take advantage of this change opportunity and improve its performance. This performance improvement can be focused on relevant stakeholder metrics that captures the utility and preferences.

First, by incorporating simulation into the system engineering process, the long list of system functional and nonfunctional requirements can be prioritized according to their ability to deliver stakeholder value at a given time epoch. This will in turn focus the operational performance (users using the system, administrators enabling the system) as well as the maintenance performance (maintainers maintaining the system, logisticians resupplying aspects of the system) of the system throughout the near term and long-term operational life cycle.

Second, by integrating simulation functionality into the system, the system has now the ability to apply simulation-based optimization 'on the fly' to support modifications that implement adaptability as well as flexibility, and—as the simulation functionality is provided where and when it is needed—ultimately agility.

Third, when the traditional system functions are replaced with agent like methods, and when the system furthermore gets equipped with the necessary sensor, perception, and communication components as described earlier for intelligent agents, the systems indeed can become antifragile by identifying unforeseen constellations, planning new options using social concepts augmented with artificial intelligence, and integrating new solution approaches into its actionable options.

In summary, the agent-based metaphor complements the simulation-based approaches to allow for taking full advantage of research results that can be mapped one to one to allow for the implementation of antifragile systems resulting from simulation-based systems engineering.

The underlying idea of the proposed efforts in this section can be summarized as follows: Using modeling and simulation, we can define a system—whether it has a real world counterpart or not—by its axioms and rules as an executable model and '*bring it to life*' as a simulation using animation and visualization. Using immersive virtual environments makes the user part of this creation. This is a powerful approach to understand things that are, that could be, or that could not be. Additionally, using established methods of artificial intelligence, as they are applied to implement intelligent software agents will also enable understanding potential future complex, adaptive system states and operational environments. These systems—virtual and real—can observe, communicate, and learn to be not only robust in preconceived environments and contexts, but to be resilient in hostile environments, and even become better (as determined by dynamic and learning utility functions) under stress.

4.4 Topics for a Research Agenda

> Methods and tools, based on **solid theoretical foundations**, will advance to address the market demands of innovation, productivity, and time to market as well as product quality and safety by harnessing the power of advancements in modeling, simulation and knowledge representation, such as domain specific standard vocabularies, thereby meeting the needs of an increasingly diverse stakeholder community. The methods and tools will also keep pace with system **complexity** that continues to be driven by customers demanding ever increasing system **interconnectedness**, **autonomy**, **ready access to information**, and other technology advances associated with the digital revolution….
>
> [A World in Motion: Systems Engineering Vision 2025, INCOSE, p. 24]

From the state of the art described in Sect. 4.2 of this chapter and the vision for a simulation-based systems engineering approach that could improve the current methods and processes, several topics for a research agenda can be derived. Simply updating the systems engineering 'V' to include simulation in all relevant processes is a first step, but not sufficient. Nonetheless, research results as captured by Morse et al. (2010) or D'Ambrogio and Durak (2016), which evaluated to what degree standard processes as captured by standardization bodies for systems engineering can be applied to simulation engineering as well, are very valuable and deserve more attention.

The obvious research topics in the context of these chapters are also addressing the integration of simulation methods and processes with systems engineering processes. A subset of related research recommendations will be given in the first subsection. Although the envisioned goal is a transdisciplinary approach (Tolk and Hughes 2014) that systematically integrates knowledge components from both domains in transcending and transgressing form, a gradual approach may ensure a smoother transition from the current state towards this ambitious end goal.

A second set of topics targets complex, adaptive solutions needed to utilize agent-based simulation in better support of systems engineering, including better support of validation and verification, management of emergence, and computational support of utility functions, which play not only a pivotal role in machine learning and decision making, they will also increasingly support human decision makers in the complex systems engineering tasks.

4.4.1 Standards for Systems Engineering and Simulation

The systems engineering community recognized the power of representation of the system itself as well as the system in its context, as described in Sect. 4.3 of this chapter. As an example, two approaches routing in the systems engineering domain shall be described here.

The Object Management Group (OMG) developed the *Foundational Subset For Executable UML Models (fUML)*. This effort identifies higher level UML concepts and the precise definition of their execution semantics. OMG (2016) summarizes the effort as follows: "In sum, the foundational subset defines a basic virtual machine for the Unified Modeling Language, and the specific abstractions supported thereon, enabling compliant models to be transformed into various executable forms for verification, integration, and deployment." The scope of what can be executed is defined by what is specified in fUML compliant UML artifacts, but usually the boundaries of the systems define also the boundaries of the executable. The simulation of the context, as described in Garcia and Tolk (2015), are not part of the specification. Similar research is conducted for the system markup language version (SysML), such as described by Kapos et al. (2014).

- The *Functional Mockup Interface (FMI)* introduced standards that allow model exchange and co-simulation in support of systems engineering activities in particular in domains that heavily depend on Computer Assisted Design (CAD) tools, originally targeting the automotive sector, but today applied in many industrial and scientific projects with a broader domain as well (Blochwitz et al. 2012). More than 30 tool providers support the standard that allows not only to use various simulation tools to simulate the intended behavior of the system and its components, it also can be used to switch between simulated components and real components, also known as 'hardware in the loop.' Again, the focus lies on the system and its components. It augments multiple existent de facto interface standards to tools like SimuLink/MathWorks, Mathematica, MATLAB, etc.

It could be argued that FMI is a simulation community effort, but a literature research quickly shows that FMI is developed and used by systems engineers who use simulation as a support tool, presenting the research results nearly exclusively to the manufacturing engineering community.

In contrast, systems engineering principles allowed for the specification of a theory of modelling and simulation by Bernie Zeigler in 1976, and was extended by him and in his research team in 2000 to reflect technical and organizational developments (Zeigler et al. 2000), becoming one of the most referenced textbooks on simulation:

- The *Discrete Event System Specification (DEVS)* is a modular and hierarchical formalism to describe the simulation of systems and its components based on how they transform a set of input parameters into output parameters. The transformation depends on the inner structure and parameters of the transforming systems and is defined as a set of time functions that map input parameters to inner parameters, inner parameters to themselves, and inner parameters to output parameters. Conflicts are resolved by an additional set of selection functions. DEVS is widely applied and considered the most used formalism to address simulation specifications. Several efforts have been conducted to show how DEVS can be used to support UML models in their execution, as demonstrated, among others, by Hong and Kim (2004) and Risco-Martin et al. (2009). Of particular interest is the work conducted by Aliyu et al. (2016) which extends the work of Shuman (2011) by defining an integrative framework for model-driven systems engineering by merging formal methods of simulation, analysis, and enactment methodologies for discrete event systems.

In the recent years, the need to support systems engineers and engineering managers with better methods and tools to understand, manage, and govern complex systems or system of systems was recognized (Gorod et al. 2008). This did lead to many recommendations that are simulation-based. An early example is Mittal et al. (2008), which is among the first publications evaluating the use of simulation in the systems of systems domain. Rainey and Tolk (2015) compiled and edited additional work on the M&S support for systems of systems engineering. A related topic is the development of net-centric systems, as potential collaboration partners are not known at the time the systems are defined and implemented. Mittal and Martin (2013) compiled methods and tools based on the DEVS Unified Process to support such challenges, highlightening the use ofmodel-driven engineering technologies, such as domain specific languages.

Unfortunately, despite all these positive cross-applications, many of such developments are still conducted in stove-pipes of the simulation and the systems engineering communities, with rare exchange of research results or requirements. Simulation books on systems engineering support are rarely mentioned in the systems engineering community and vice versa. A recent workshop conducted by the National Science Foundation (Fujimoto 2016) hopefully marks a clear next step to break down these stove-piped approaches and lead to an increase in inter- and transdisciplinary research. The alignment of data and harmonization of processes needs to be in the focus of research to establish a common foundation.

Additional topics requiring research are the technologies allowing for immersive virtual environments. As high-resolution visualization requires to present data in the most efficient form in support of minimizing computational resources needed for the transformation, many immersive visualization solutions use proprietary formats and methods, that often are the results of intensive and expensive internal research effort. Nonetheless, the definition of standardized interfaces to seamlessly integrate systems engineering artifacts, their simulation, and their visualization in the context of their situated virtual representation of the operational environment must be an objective to maximize the synergy of inter- and transdisciplinary research.

4.4.2 Improving Complex Adaptive Solutions

The second set of ideas proposed in this chapter, aimed at significantly improving the contributions of simulation-based systems engineering, considers replacing the well-defined system functions with *self-modifying methods*, as motivated by methods used within intelligent software agents. There are many research topics perceivable to improve creating a perception, better integration of sensors, efficient access to memory of important decisions and associated consequences to support better decision making and learning, etc. However, there are two topics central to this chapter, and as such require additional evaluation.

- The *validation of agent-based solutions* is a topic of general interest in the agent-based simulation community. Validation ensures the accuracy of representation of the domain of interest. In other words answering the question: "Did we model the correct thing?" In particular when the behavior of an agent can change, due to learning or adaptation to a new set of requirements or a change in the environment, how can this unforeseeable behavior be validated? If empirical data are available, at least the comparison of some degree of similarity between expected and observed behavior is possible, but a scientifically accepted method has not yet been generally accepted. Of particular interest are systems that expose some kind of emergent behavior. The work of Szabo and Tao (2013) provides first ideas on how to use semantic technologies to validate such emergence in component-based systems. A more traditional approach has been described by Gore et al. (2016). They are extending the ideas of statistical debugging for the support of validation of simulation systems, including agent-based simulation. While their approach does not validate the system, it traces the behavior of the simulation and protocols if a set of assumptions and constraints is always satisfied for these observed core parameters.
- Although utility theory is well established in support of rational decision-making, the free *machine-based definition of utility functions* to capture when and which new objectives are needed has not been addressed sufficiently so far. Afriat's (1967) foundational work showed how empirical data can be used to define utility function in hindsight, but can this method be applied to

agile systems? The data describing such new challenges may not even have been included into the originally defined data set of important parameters to observe. The question when and how we decide that a change of decision parameters is needed is unanswered as well. Even human decision makers have problems in recognizing that a new paradigm is emerging, so how can this be recognized by a system? Simple brute-force approaches generating randomly new utility functions that may better work in a new environment only work in very simple toy worlds, as the solution space in real world problems is huge and complex, resulting quickly in computational infeasibility of such approaches. Heuristics like genetic algorithms showed some promise, but they are limited by the parameters that describe their solution space, which brings us back to the problem to understand which parameters are important at the first place. De Florio (2014) provides related ideas allowing the utilization of machine learning to enable comparable capabilities.

- The architecting of complex, adaptive systems may require an extension of the currently applied systems engineering practices, leading to an improved set of *system engineering principles for adaptive and antifragile systems*. Pitsko (2014) already recommends first principles for architecting adaptable complex systems, and Jones (2014) addresses respective aspects for antifragile systems engineering. These new ideas have to be aligned and integrated into the body of knowledge.

As in the earlier observations, research contributions are made in several communities that focus on their domain-specific solutions and do not always share the results beyond their boundaries. The number of relevant disciplines is even growing for the research questions proposed here. Autonomous systems have a strong topological similarity to intelligent software agents, which motivates the use of agent-based method to test rulesets to be used in autonomous systems (Tolk 2015). Big data and deep learning has the potential to enhance—or maybe even replace—utility theory based approach in the traditional sense. How can these results help research in the context defined in this chapter?

The topics captured in this section are neither complete nor exclusive, but the authors hope they will help to ignite a multidisciplinary discussion that will lead to inter- and transdisciplinary research by increased exchange of information, and ultimately to the definition of a transdisciplinary methods for simulation-based systems engineering.

4.5 Discussion and Summary

Within this section, the authors showed that the state of the art of systems engineering already uses simulation methods in support of several of their processes. Organizations well known for their systems engineering expertise are actively promoting the use of simulation. The National Defense Industrial Association

(NDIA) established its Simulation Committee within the Systems Engineering Department. MITRE declares on its websites that "one of our goals is to find ways to reduce system development time and costs through simulations, which are faster and less expensive than testing new systems in the field" (MITRE 2017).

However, the integrated use of simulation methods as the *standard way* to evaluate numerically the dynamic behavior of systems—internally as well as its context—is limited to often still to be better aligned activities. The Body of Knowledge of Systems Engineering (BKCASE 2016) in its last version mentions simulation as a tool in many of its sections, but without sufficient specifics or references to the work addressed in this chapter. While the role of modeling and model-based engineering has own sections to cope with challenges and recommended good practices, a section coping with simulation-based systems engineering topics is missing. While many application motivations and examples are given, the step recognition of simulation not only as a tool, but as a main topic that deserves its own section to provide access to the relevant knowledge within a section of this archival documentation has not yet been made. A positive example is the related Body of Knowledge of Engineering Management which recognizes simulation as its only method, as described by Tolk et al. (2009).

What is generally needed for a real leap in quality and efficiency is to move from individual multidisciplinary efforts over interdisciplinary alignments to a real *new transdisciplinary reality*, as discussed by Stock and Burton (2011), in which systems engineering and modeling and simulation become a new discipline that not only systematically integrate the body of knowledge of both domains, but that will create new knowledge elements of great benefit to the users.

The NDIA Study on model-based engineering (Bergenthal 2011) identifies many support opportunities and gives examples and use cases, but these examples are not used yet to established a common view on how to use modeling and simulation systematically and systemically to transform systems engineering as envisioned in this chapter. Glazner (2011) describes a framework that allows various simulation tools to come together to support the dynamic analysis enterprise architectures for potential performance. The same approach is applicable systems engineering processes that are orchestrated by a common system architecture as well. The general challenges of using hybrid simulation approaches have been recently addressed by Powell and Mustafee (2014). To ensure the applicability of such results in all affected communities, the communication and alignment of such results is pivotal. Transdisciplinary simulation-based systems engineering is already supported by technical solutions, now the communities must be aligned better in their activities.

Such a transdisciplinary approach will also facilitate the second topic presented in this chapter that may elevate systems engineering to the next level: Integrating the research in the domain of intelligent software agents and agent-directed simulation for systems engineering (Yilmaz and Ören 2009) into this domain, allowing systems to become social, observe their environment, evaluate the effects of their action, and learn from these experiences. Such efforts promise to increase the 'ilities' of systems significantly, including the development of antifragile systems, which get better under stress (Taleb 2012).

Several research topics have been identified that hopefully will allow for increased interdisciplinary research as the next step to better understand complexity and emergence in our increasingly interconnected world to ensure that simulation-based systems engineering will become one of the most influential disciplines in the near future. The technical framework and many building blocks to allow for their development already have been developed, but now they need to be integrated and published and socialized in both communities. The authors hope that this chapter contributes to make the vision of real synergy between simulation and systems engineering, as among others described by Ören and Yilmaz (2006) and Gianni et al. (2014), a reality, not only in theory but also for practitioners in the field.

Review questions

1. What are the challenges with using traditional simulation and optimization approaches when analyzing system performance, and when might you want to optimize around a point solution, and when would you not? What are the benefits and limitations of high fidelity point analysis in the context of the larger system and environment?

2. What are the differences between a situated environment and an immersive environment in their ability to support simulation-based systems engineering? What role do executive architectures play in this context? How can this be used to integrate the user early into system specification and testing?

3. How would you suggest incorporating expected and unexpected environmental dynamics into a systems engineering project? Particularly interesting are antifragile systems that get better under such stress. How can the use of agent-based methods help to create such antifragile systems using simulation-based systems engineering?

4. How can machine learning be implemented when machine understanding is realized using pre-programmed meta-models that represent the knowledge of the system environment at the moment of the system installation? How can a systems engineering environment and approach be designed to allow for the development of a system, which can intelligently adapt?

5. What role have standards played in advancing simulation? Where are standards most needed to facilitate the adoption of simulation-based system engineering? How could the incorporation of standards enable integration of executable architectures and simulation environments?

6. How can simulation make its way into the toolbox of systems engineers and into the context of model-based systems engineering as a general solution method, as opposed to merely as a set of tools?

References

Afriat, S. N. (1967). “The construction of utility functions from expenditure data.” International economic review, 8(1), 67–77.

Aliyu, H. O., Maïga, O., & Traoré, M. K. (2016). “The high level language for system specification: A model-driven approach to systems engineering.” International Journal of Modeling, Simulation, and Scientific Computing, 7(01), 1641003.

Bergenthal, J. (2011). Final Report Model Based Engineering (MBE) Subcommittee, National Defense Industrial Association, Arlington, VA.

BKCASE Editorial Board. (2016). The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.7. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology: www.sebokwiki.org [last accessed February 2017].

Blochwitz, T., Otter, M., Akesson, J., Arnold, M., Clauß, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., Olsson, H., & Viel, A. (2012). “Functional mockup interface 2.0: The standard for tool independent exchange of simulation models.” Proc. 9th International MODELICA Conference; Linköping University Electronic Press, pp. 173–184.

Booher, H. R. (2003). Handbook of human systems integration (Vol. 23). John Wiley & Sons.

Brill, J. H. (1998). “Systems engineering— A retrospective view.” Syst. Engin., 1: 258–266. doi:10.1002/(SICI)1520-6858(1998).

Campbell, D., Wampole, G., & Wheeler, T. (2015). “Using Model Based Engineering to Own the Technical Baseline.” Procedia Manufacturing, 3:1995–2002, ISSN 2351-9789.

D’Ambrogio, A., & Durak, U. (2016). “Setting systems and simulation life cycle processes side by side.” International Symposium on Systems Engineering (ISSE), IEEE Press, 128–134.

De Florio, V. (2014). “Antifragility = elasticity + resilience + machine learning models and algorithms for open system fidelity.” Procedia Computer Science 32, 834–841.

De Weck, O. L., Ross, A. M., & Rhodes, D. H. (2012). Investigating relationships and semantic sets amongst system lifecycle properties (Ilities). Engineering Systems Division Working Paper ESD-WP-2012-12, MIT Press, Cambridge, MA.

Deihl, E. & Sterman, J. (1995). “Effects of feedback complexity on dynamic decision making.” Organizational Behavior and Human Decision Processes 62(2), 198–215.

Estefan, J. A. (2007). “Survey of model-based systems engineering (MBSE) methodologies.” Report of the Incose MBSE Focus Group 25.8.

Forrester, J. W. (1961). Industrial Dynamics. Pegasus Communications, Waltham, MA.

Fujimoto, R. (2016). Research Challenges in Modeling & Simulation for Engineering Complex Systems. NSF Report, Arlington, VA, accessible via http://trainingsystems.org/publications/Research-Challenges-in-Modeling-and-Simulation-for-Engineering-Complex-Systems.pdf [last accessed February 2017].

Garcia, J. J. & Tolk, A. (2015). “Executable Architectures in Executable Context enabling Fit-for-Purpose and Portfolio Assessment” Journal for Defense Modeling and Simulation 12(2):91–107.

Gianni, D., D’Ambrogio, A., & Tolk, A. (Eds.). (2014). Modeling and Simulation-Based Systems Engineering Handbook. CRC Press.

Glazner, C. G. (2011) “Enterprise transformation using a simulation of enterprise architecture.” Journal of Enterprise Transformation 1(3): 231–260.

Gore, R. J., Lynch, C. J., & Kavak, H. (2016). “Applying statistical debugging for enhanced trace validation of agent-based models.” Simulation, DOI 10.1177/0037549716659707.

Gorod, A., Sauser, B. & Boardman, J. (2008). “System-of-systems engineering management: A review of modern history and a path forward.” IEEE Systems Journal, 2(4): 484–499.

Griffin, M. D. (2010). “How do we fix systems engineering?.” Proc. 61st Annual International Congress, Prague, Czech Republic. Vol. 27.

Haskins, C. (2011). “A historical perspective of MBSE with a view to the future.” Incose International Symposium, 21(1).

Holder, B. & Wallace, M. (1998). Lockheed-Martin F-22 Raptor: An Illustrated History. Schiffer Publishing, Series Military/Aviation History. Atglen, PA. ISBN 978-0-76430-558-0.

Hong, S. & Kim, T. G. (2004). "Embedding UML subset into object-oriented DEVS modeling process." SIMULATION SERIES, 36(4), 161.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). "A roadmap of agent research and development." Autonomous Agents and Multi-Agent Systems, 1(1):7–38.

Kapos, G. D., Dalakas, V., Tsadimas, A., Nikolaidou, M., & Anagnostopoulos, D. (2014): "Model-based system engineering using SysML: Deriving executable simulation models with QVT," 8th Annual IEEE Systems Conference (SysCon), 531–538, IEEE Press.

Jones, K. H. (2014). "Engineering antifragile systems: A change in design philosophy." Procedia computer science 32:870–875.

Kranz, G. (2001). "Failure is not an option: Mission control from Mercury to Apollo 13 and beyond." Simon and Schuster.

McCullough, D. (2015). The Wright brothers: The dramatic story-behind-the-story. Simon and Schuster, 2015.

MITRE (2017) https://www.mitre.org/research/core-technology-research/modeling-simulation, [last accessed February 2017].

Mittal, S. (2006). Extending DoDAF to allow integrated DEVS-based modeling and simulation. The Journal of Defense Modeling and Simulation, 3(2), 95–123.

Mittal, S. & Martín, J. L. R. (2013). Netcentric system of systems engineering with DEVS unified process. CRC Press.

Mittal, S., Zeigler, B. P., Martín, J. L. R., Sahin, F., & Jamshidi, M. (2008). "Modeling and simulation for systems of systems engineering." In Systems of Systems–Innovations for the 21st Century, edited by M. Jamshidi, John Wiley & Sons, pp. 101–149.

Morse, K. L., Coolahan, J., Lutz, B., Horner, N., Vick, S., & Syring, R. (2010). Best practices for the development of models and simulations. Johns Hopkins University-Applied Physics Laboratory, Laurel, MD, USA.

Object Management Group (2016). Semantics of a Foundational Subset for Executable UML Models (fUML), Version 1.2.1. OMG Document Number: 2016-01-05.

Ören, T. (1984a). "Model-based activities: A paradigm shift." In Simulation and model-based methodologies: an integrative view, edited by T. Ören, B. P. Zeigler, & M. S. Elzas. Springer Berlin Heidelberg, pp. 3–40.

Ören, T. I. (1984b). "GEST—a modelling and simulation language based on system theoretic concepts." In Simulation and model-based methodologies: an integrative view, edited by T. Ören, B. P. Zeigler, & M. S. Elzas. Springer Berlin Heidelberg, pp. 281–335.

Ören, T. I. & Yilmaz, L. (2006). "Synergy of systems engineering and modeling and simulation." Proceedings of the 2006 International Conference on Modeling and Simulation-Methodology, Tools, Software Applications (M&S MTSA), pp. 10–17.

Pawlowski, T., Barr, P. C., & Ring, S. J. (2004). Applying Executable Architectures to Support Dynamic Analysis of C2 Systems. Technical Paper, Mitre Corporation, McLean, VA.

Pistko, R. (2014). Principles for Architecting Inherently Adaptable Complex Systems. Ph.D. Dissertation at the Stevens Institute of technology, Hoboken, NJ.

Powell, J. & Mustafee, N. (2014). "Soft OR Approaches in Problem Formulations Stage of a Hybrid M&S Study." Proceedings of the 2014 Winter Simulation Conference, IEEE Press, 1664–1675.

Rainey, L. B. & Tolk, A. (Eds.). (2015). Modeling and simulation support for system of systems engineering applications. John Wiley & Sons.

Risco-Martín, J. L., De La Cruz, J. M., Mittal, S., & Zeigler, B. P. (2009). "eUDEVS: Executable UML with DEVS theory of modeling and simulation." Simulation, 85(11–12), 750–777.

Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2008). Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. Systems Engineering, 11(3), 246–262.

Shuman IV., E. A. (2011). Understanding the Elements of Executable Architectures through a Multi-Dimensional Analysis Framework. Ph.D. Dissertation, Batten College of Engineering, Old Dominion University, Norfolk, VA.

Smith, E. C. (1962). "Simulation in systems engineering." IBM Systems Journal, 1(1), 33–50.

Sterman, J. (1989). "Misperceptions of feedback in dynamic decision making." Organizational Behavior and Human Decision Processes 43(3), 301–335.

Sterman, J. (1992). "Teaching takes off: flight simulators for management education." OR/MS Today, 35(3):40-44.

Stock, P. & Burton, R. J. F. (2011). "Defining Terms For Integrated Multi-Inter-Trans-Disciplinary Sustainability Research." Sustainability 3:1090–1113; doi:10.3390/su3081090.

Szabo, C. & Tao, Y. M. (2013). "Semantic validation of emergent properties in component-based simulation models." in Ontology, Epistemology, and Teleology for Modeling and Simulation, edited by A. Tolk, Springer, Berlin Heidelberg, pp. 319–333.

Taleb, N. N. (2012). Antifragile: Things that gain from Disorder. Random House Inc., New York NY.

Tolk, A. & Hughes, T.K. (2014). "Systems engineering, architecture, and simulation," in Modeling and Simulation-Based Systems Engineering Handbook, edited by D. Gianni, A. D'Ambrogio, & A. Tolk, CRC, pp. 11–41.

Tolk, A. & Uhrmacher, A. M. (2009). "Agents: Agenthood, Agent Architectures, and Agent Taxonomies," in Agent-Directed Simulation and Systems Engineering, edited by L. Yilmaz & T. Ören, Wiley, Berlin Germany, pp. 75–109.

Tolk, A. (2015). "Merging two worlds: agent-based simulation methods for autonomous systems." in Autonomous Systems: Issues for Defence Policymakers. NATO Headquarters Supreme Allied Commander Transformation, Norfolk, pp. 289–315.

Tolk, A., Rabadi, G., & Merino, D. N. (2009). "Embedding simulation education into the engineering management body of knowledge." International Journal of Simulation and Process Modelling 5(1), 14–19.

van Cleave, D. A. (2010). "Model-Driven Engineering Allows New Design Methods to Improve an Existing Aircraft System" MITRE Corporation News Article; accessible at https://www.mitre.org/publications/project-stories/modeldriven-engineering-allows-new-design-methods-to-improve-an-existing-aircraft-system [last accessed February 2017].

Wagenaar, W. & Timmers, H. (1978). "Extrapolation of exponential time series in so enhanced by having more data points." Perception and Psychophysics, 24:182–184.

Wagenaar, W. & Timmers, H. (1979). "The pond-and-duckweed problem: Three experiments in the misperception of exponential growth." Acta Psychologica, 43:239–251.

Wagenhals, L. W., Liles, S. W., & Levis, A. H. (2009). "Toward executable architectures to support evaluation." Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS'09), IEEE Press, pp. 502–511.

Wheeler, T. M., & Brooks, M. D. (2007). Experiences in Applying Architecture-Centric Model Based System Engineering to Large-Scale, Distributed, Real-Time Systems. Technical Paper, Mitre Corporation, McLean VA.

Wymore, A. W. (1967). A mathematical theory of systems engineering: the elements. New York: John Wiley & Sons.

Wymore, A. W. (1993). Model-based systems engineering (Vol. 3). CRC press.

Yilmaz, L. & Ören T. (2009). Agent-Directed Simulation and Systems Engineering. Wiley, Berlin Germany.

Zeigler, B. P. (1986). "Toward a Simulation Methodology for Variable Structure Modeling." in Modeling and Simulation Methodology in the Artificial Intelligence Era, edited by M. S. Elzas, T. Ören, & B. P. Zeigler, North Holland, Amsterdam, pp. 195–210.

Zeigler, B. P., Praehofer, H., & Kim, T. G. (2000). Theory of Modeling and Simulation: Integrating discrete event and continuous complex Dynamic Systems. Academic Press.

Zyda, M. (2005). "From visual simulation to virtual reality to games." Computer 38(90: 25–32.

Author Biographies

Andreas Tolk is Technology Integrator for the Modeling, Simulation, Experimentation, and Analytics Division of The MITRE Corporation, and an Adjunct Full Professor at Old Dominion University. He holds a Ph.D. and M.S. in Computer Science from the University of the Federal Armed Forces in Munich, Germany. He is a Fellow of the Society for Modeling and Simulation (SCS), and a Senior Member of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). He edited seven books on simulation and systems engineering topics and published more than 300 journal articles, book chapters, and conference papers. He received the Frank Batten Excellence in Research Award from Old Dominion University, the Technical Merit Award by the Simulation Interoperability Standards Organization (SISO), and the Outstanding Professional Contribution and Distinguished Professional Achievement Awards from SCS. His research interest focuses on enablers for transdisciplinary systems engineering and modeling and simulation, including their epistemological, mathematical, and computational foundations and constraints.

Christopher G. Glazner is the Department Head for Model-based Analytics in the Modeling, Simulation, Experimentation and Analytics Division of The MITRE Corporation. He holds a Ph.D. in Engineering Systems and an MS in Technology Policy from the Massachusetts Institute of Technology, and a BS in Electrical Engineering and a BA in Plan II from the University of Texas at Austin. At MITRE, he leads research and direct work in support of the government that employs data, modeling, simulation, and visualization to understand and engineer complex, socio-technical systems. His research interests include immersive visualization for modeling, human-model interaction, and architectural simulation. He served as an editor on the "The Enterprise Dynamics Sourcebook."

Robert Pitsko is the Department Head of the End to End Systems Engineering Department of the Systems Engineering Technical Center for the MITRE Corporation. He was pivotal in initiating the MITRE Innovation Program Agile Enterprises research portfolio, which funds over 20 active research projects. He received the M.E. and Ph.D. in Systems Engineering from Stevens Institute of Technology and his B.S. in Computer Science from the Pennsylvania State University. As an Adjunct Professor with Stevens Institute of Technology, he teaches the graduate courses on the Foundation of Systems Engineering and System Architecture and Design. As a Doctoral Fellow of the Systems Engineering Research Center (SERC) he routinely provides talks describing Complex Systems research to the academic and DoD sponsor community. His professional and research activities emphasize systems engineering and design with a focus on conceptual design evaluation, adaptability, complex system design, system architecture, and System of Systems Engineering.

Chapter 5
Simulation-Based Cyber-Physical Systems and Internet-of-Things

Bo Hu Li, Lin Zhang, Tan Li, Ting Yu Lin and Jin Cui

Abstract Based on related research of the authors' team, this chapter gives a full picture of simulation-based cyber-physical systems (SB-CPS) and simulation-based internet-of-things (SB-IoT). Definitions and explanations of the concepts of CPS/IoT and SB-CPS/SB-IoT are introduced. Technical challenge from CPS and IoT and challenge of M&S technology in SB-CPS/SB-IoT are analyzed. Body of knowledge/technology of the SB-CPS/SB-IoT is proposed. Key technologies enabling the SB-CPS/SB-IoT are described. These technologies include SB-CPS/SB-IoT modeling theory and method, SB-CPS/SB-IoT simulation system theory and technology, SB-CPS/SB-IoT simulation application engineering theory and technology. Furthermore, the impact of SB-CPS/SB-IoT on society and economy, people's livelihood and national security are discussed. Some application cases, e.g., smart cities, smart manufacturing, are illustrated. Finally, some suggestions for future works are given.

Keywords Big data-based modeling · Body of knowledge · Complex hybrid network modeling · Complex hybrid network system · Complex system modeling · Cyber factory · Deep learning modeling · Embedded simulation system · High-performance parallel algorithm · Intelligent manufacturing · Internet-of-things (IoT) · Simulation of smart cities · Simulation-based cyber-physical systems · Smart cloud simulation system · Time-critical system modeling

B.H. Li · L. Zhang (✉) · J. Cui
School of Automatic Science and Electrical Engineering, Beihang University (Beijing University of Aeronautics and Astronautics), Beijing, China
e-mail: zhanglin@buaa.edu.cn

B.H. Li · T. Li · T.Y. Lin
Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing, China

B.H. Li · T. Li · T.Y. Lin
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing, China

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_5

5.1 Introduction

5.1.1 Definitions and Explanations of the Concepts of CPS/IoT

The continuous evolution of computing and networking technologies is creating a new world populated by many sensors on physical and social environments. This emerging new world goes much further than the original visions of ubiquitous computing and World Wide Web. Aspects of this new world have received various names such as Cyber-Physical Systems (CPS) and Internet-of-Things (IoT). Both CPS and IoT are typical complex systems.

(1) CPS

The concept of CPS was first proposed in 2006. However, experts and scholars from different institutes or countries focus on different emphasis of CPS. Considering existing research results comprehensively, we defined the existing CPS as a class of complex systems that realize integration optimization operations under given objectives as well as temporal and spatial constraints to seamlessly integrate the cyberspace and physical space for state perception, real-time analysis, scientific decision-making and accurate execution of human, machine, material, environment and information sectors autonomously and intelligently, by employing advanced computing and communication technology, automatic control technology as well as data-driven technology. Figure 5.1 shows that, CPS consists of three levels, including unit level, system level and system-of-system (SoS) level.

(2) IoT

The terminology "Internet-of-Things" was originated in 1999. At present, the vision of the International Telecommunication Union (ITU) is generally accepted. "from anytime, anyplace connectivity for anyone, we will now have connectivity for anything", "Through the exploitation of identification, data capture (by devices of radio frequency, infrared, optical and galvanic driving), processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled." (ITU 2005; ITU-T 2012) The system architecture of IoT is shown in Fig. 5.2.

(3) The relationship between IoT and CPS

IoT and CPS share many core technology elements. Both CPS and IoT are networked systems and likely to involve both aspects of the physical and cyber worlds. IoT is the basic infrastructure and can be considered as a smart cyberspace providing ubiquitous connectivity to smart computational devices and mobility via Internet worldwide. Meanwhile, CPS can be considered as IoT-enabled. CPS links many physical sensor data to detailed simulation models running on large data centers. IoT brings together many appliances, making much more environmental data available and supporting control of these appliances (Pu 2011).

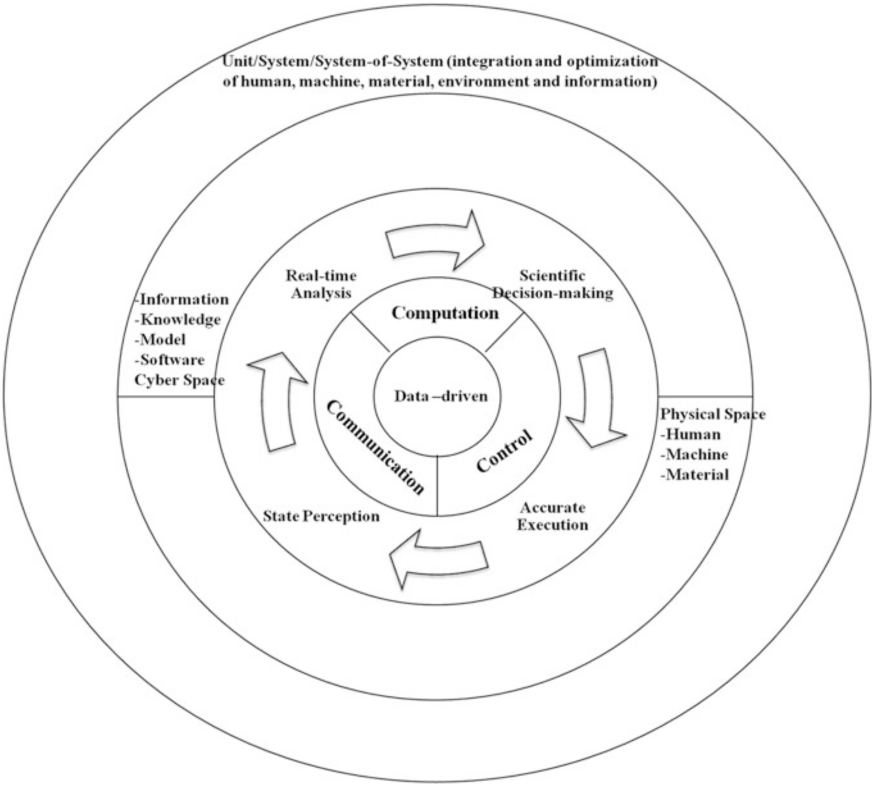


Fig. 5.1 System architecture of CPS

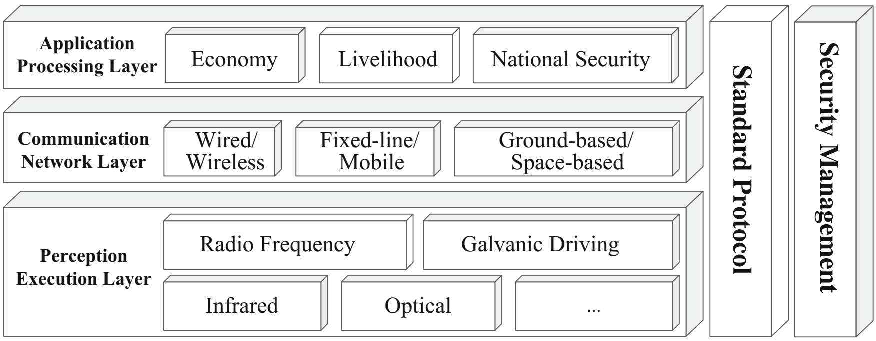


Fig. 5.2 System architecture of IoT

Human, machine, material, environment and information sectors in IoT systems have different connection relationships like open-loop connection and closed-loop connection, serial connection and parallel connection, as well as iteration connection. CPS is a critical type of IoT. Human, machine, material, environment and information sectors in CPS systems need to form a closed-loop feedback control connection relationship. Generally speaking, CPS is the evolved form of IoT and it has the potential to become the next generation intelligent system which integrating computation, communication, and control.

IoT/CPS is widely used in the fields of economy, society, and national security. The typical systems of IoT/CPS includes the complex engineering system such as the aeronautics and astronautics system, the complex social system such as economic planning and city system, the complex biological system such as human, animal and plant system, the complex environment system such as climate and electromagnetic system, the complex military system such as Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) system, the complex network system such as ubiquitous network system.

5.1.2 Connotation of SB-IoT and SB-CPS

According to the above discussion, IoT/CPS is a complex system composed of "hardware" (such as sensor and automatic controller), "software" (such as industrial software), "network" (such as industrial network) and "platform" (such as industrial cloud platform) (Chinese CPS Development Forum 2017). Especially, the IoT/CPS of system level and SoS level is a kind of complex system that system composition and architecture are complex, system mechanism is complex, the interactions and energy exchanges between subsystems or between the system and its surroundings are complex, moreover, the overall properties of system are featured with variable-structured, nonlinear, self-organization, emergence, chaotic and gaming, etc.

Therefore, M&S technology is a very important and effective means. It could not be limited by time and space to observe and study phenomena that have happened or have not happened, and the occurrence and development process of these phenomena under various imaginary conditions. It can help people go deep into the macro or micro-world where science and human could not reach, which provides a new method and means for the human understanding and transforming the world, greatly expanding the ability of human to understand the world. This is the background of the Simulation-based CPS/IoT proposed in this chapter.

Simulation-based CPS/IoT means modeling and simulation technology get involved through the whole life cycle activities of CPS and IoT systems, including augmentation, design, experiment, production, operation, management, and service. In particular, modern modeling and simulation system technology plays an important role in supporting the operation activity of the whole CPS or IoT system which could be regarded as embedded simulation, and it is a significant kind of SB-CPS/SB/IoT. The architectures of SB-CPS and SB-IoT are presented in Figs. 5.3 and 5.4.

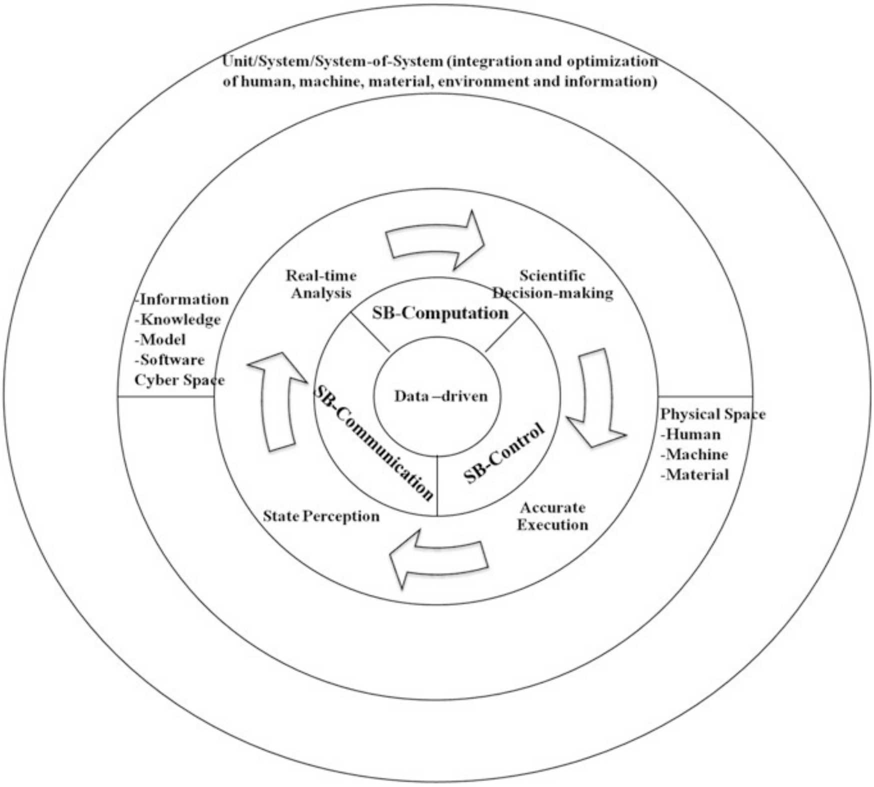


Fig. 5.3 System architecture of SB-CPS

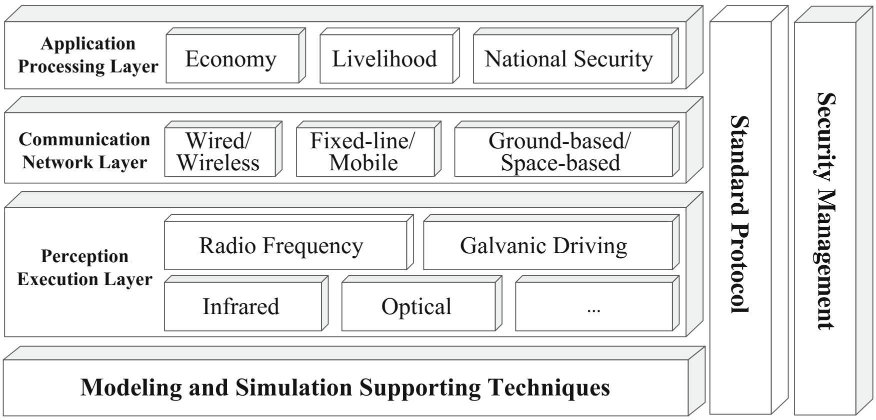


Fig. 5.4 System architecture of SB-IoT

5.1.3 Outline of This Chapter

This chapter gives a full picture of SB-CPS and SB-IoT. Based on the Definitions and explanations of the concepts of CPS/IoT and SB-CPS/SB-IoT, technological architectures of the SB-CPS and SB-IoT are proposed. Key technologies enabling the SB-CPS and SB-IoT are described. These technologies include advanced modeling theory and techniques, advanced modeling and simulation system techniques, advanced modeling and simulation engineering techniques, etc. Furthermore, the impact of SB-CPS and SB-IoT on society and economy, people's livelihood and national security are discussed. Some application cases, e.g., smart factory, intelligent traffic and cloud manufacturing, are illustrated. Finally, some suggestions for future works are given.

5.2 Challenge of the SB-CPS/SB-IoT

5.2.1 Technical Challenge from CPS and IoT

CPS/IoT is a typical complex system, which brings great challenges to technical research, including, (1) Overall Sensor Technology, which utilizes sensor technologies to obtain information of things anytime anywhere; (2) Reliable Transmission Technology, which transmits information of things properly and precisely; (3) Intelligent Process Technology, which utilizes intelligent science technology to analysis and process the massive data and information to achieve intelligent control.

5.2.2 Challenge of M&S Technology in SB-CPS/SB-IoT

SB-CPS/SB-IoT raises great challenges to M&S Technology in modeling theory and technology, modeling and simulation system technology, and simulation application engineering technology, including,

5.2.2.1 Challenges in Modeling Theory and Technology

(1) Complex Hybrid Network System Modeling and Simulation. Most CPS/IoT networks are hybrid networks, which are composed of high-speed stable industry intranet, low-speed high-delay internet/mobile Internet, as well as the high-bandwidth low through-put space-based network.

(2) Quantitative and Qualitative Hybrid System Modeling. The kernel target of CPS is to realize the fusion of Human-Machine-Things. The existence of human qualitative process feature, the quantitative and qualitative Hybrid system modeling is required in SB-CPS/SB-IoT.
(3) Big Data based Modeling. Data generated from CPS/IoT are featured with big Volume (PB/EB), great variety (structured/unstructured data), high velocity (fast-generating) and low value intensity. Big Data based Modeling becomes significant modeling method in SB-CPS/SB-IoT.
(4) Artificial Intelligence (AI) based Mining and Modeling. Most CPS/IoT systems are featured with uncertainty and high complexity, which result poor ability in modeling CPS/IoT by analytic methods. AI-based mining and modeling methods, like Artificial Neural Network, Genetic Algorithm, Deep Learning, are required.

5.2.2.2 Challenges in Modeling and Simulation System Technology

(1) Hyper-performance real-time simulation computation. Most CPS/IoT equipment are real-time on-board equipment, whose clock period of data generation and action response is closed to or even faster than the computer clock period, which requires the hyper-performance real-time computation.
(2) High performance, high bandwidth, and low-latency synchronization/communication network. Massive (Billions of) sensors and Operational Technology (OT) equipment are integrated in CPS/IoT, which requires ability in Massive Network Coding and Location Tracing.
(3) Edge Simulation. Most simulation computation in CPS/IoT is required to execute in the frontier equipment-end to ensure the real-time ability, as well as the collaborative simulation ability between high-performance simulation and low-performance embedded computation.
(4) Cloud Simulation. CPS/IoT is featured with distribution and heterogeneous nature, which requires effective integration and collaboration with distributed heterogeneous models.
(5) High-performance Simulation. High-performance Cluster (HPC) has become important infrastructure for SB-CPS/SB-IoT to accomplish high-speed real-time simulation. Thus, the specific hardware/software for HPC-based simulation is required.

5.2.2.3 Challenges in Simulation Application Engineering Technology

(1) Prediction Simulation. In CPS/IoT, simulation-based real-time prediction of overall system behavior pattern and performance is required.

(2) VV&A (Verification, Validation and Accreditation). To ensure the correction of the results of SB-CPS/SB-IoT, VV&A methods are required, including the VV&A of system models, simulation models/systems and simulation executions.
(3) VR/AR (Virtual Reality/Augmented Reality) simulation results display. The results from SB-CPS/SB-IoT are extremely complicated, as well as strong constrains with on-board equipment. VR/AR can visualize the massive result data in user-friendly way.

5.3 Body of Knowledge/Technology of the SB-CPS/SB-IoT

The body of knowledge/technology of SB-CPS/SB-IoT includes SB-CPS/SB-IoT modeling theory and method, simulation system theory and technology, simulation application engineering theory and technology, shown as follows (Fig. 5.5).

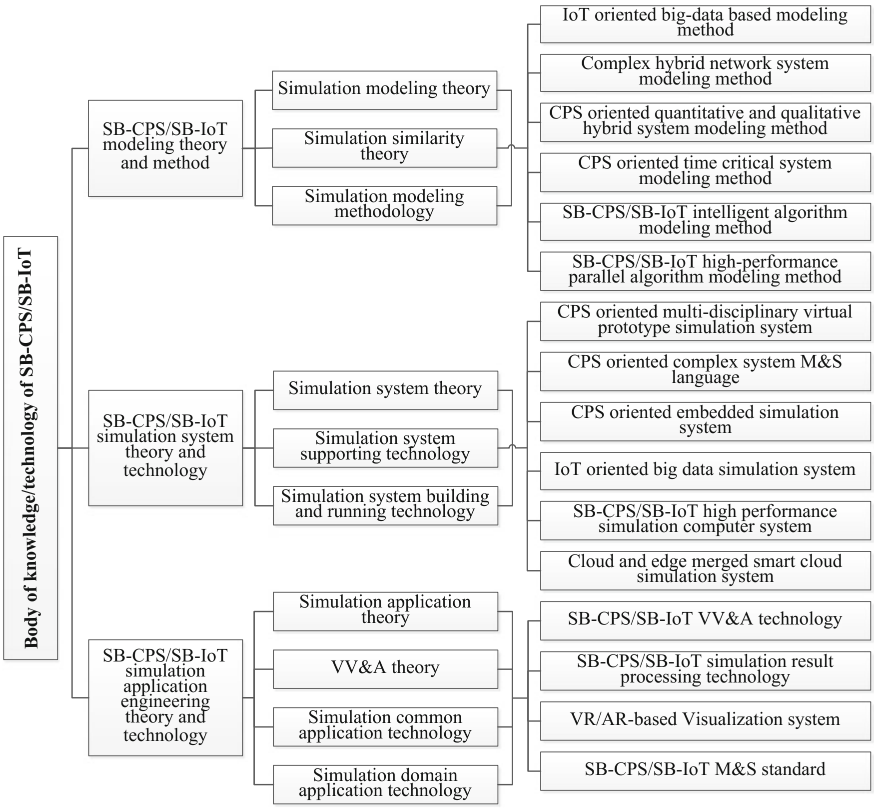


Fig. 5.5 Body of Knowledge/technology of the SB-CPS/SB-IoT

5.4 Key Technologies Enabling SB-CPS/SB-IoT

5.4.1 SB-CPS/SB-IoT Modeling Theory and Method

5.4.1.1 Big Data-Based Modeling

Big Data based Modeling Method is a type of simulation modeling method which utilizes the massive observation and application data to setup the model of complex systems, including,

(1) Big Data based reverse design of data: a kind of modeling method based on a large number of observation data to the product/system configuration, such as the scanning of the car shape data, to find the corresponding fitting model to reverse design.
(2) Big Data based Neural network training and modeling: learning and training of artificial neural network model through massive application of data, the formation of executable simulation model.
(3) Modeling based on big data clustering analysis: cluster analysis based on the model of the system behavior, the embedded model has become the system model parameters.

5.4.1.2 Complex Hybrid Network Modeling Methods

Most CPS/IoT networks are typical hybrid networks, which are composed by high-speed stable industry intranet, low-speed high-delay internet/mobile internet, as well as the high bandwidth low through-put sky-based network. Complex Hybrid Network Modeling Methods mainly research on: (1) topology structure modeling (mini-world model, no vector model, etc.); (2) statics features modeling (clustering coefficient and degree distribution modeling); (3) dynamic evolution mechanism modeling (network synchronization, nonlinear dynamic complex network) (Liu et al. 2005).

5.4.1.3 Complex System Modeling

Research on simulation modeling method for qualitative and quantitative mixed system basically includes three aspects

(1) Qualitative and quantitative Unified Modeling Method, including the system top-level description that is responsible for the top-level description of the static structure and dynamic behavior of the system, and the domain-oriented description that is responsible for the description of kinds of domain models (including quantitative, qualitative models). The research fruit of our team is QR (quantitative-rule)-QA (quantitative-Agent) modeling method (Fan et al. 2009).

(2) Modeling the interface of quantitative and qualitative interaction: Converting the quantitative and qualitative interaction data into specific structure and format needed by the qualitative model and the quantitative model. The research fruits of our team in this part are the Quantitative and Qualitative Simulation Mark-up Language (QQSRML) (Fan et al. 2009), and the Fuzzy Casual Directed Graph (FuzzyCDG) (Li et al. 2011a).
(3) Qualitative and quantitative time advance mechanism. The research fruit of our team is a QR (quantitative-rule)-QA (quantitative-Agent) mixed time advancing method, which realizes qualitative/quantitative mixed, hierarchical computing control and management of heterogeneous models (Fan et al. 2009).

Simulation modeling method based on meta-modeling framework

(1) Simulation modeling method based on meta-modeling framework mainly researches on the meta-model-based unified simulation modeling of the complex system features like multi-disciplinary, heterogeneous, and emergence.
(2) The research fruit of our team is Meta-Modeling Framework (M2F) (Li et al. 2011a, b), which proposes a hierarchical meta-model architecture, separating the continuous, discrete, qualitative mixed heterogeneous system models at the abstract level theoretically, so as to achieve unified top-level modeling for complex systems.

Simulation modeling method for variable structure system

(1) Research of simulation modeling method for variable structure system mainly focuses on the dynamic variability of the simulation model content, interface, and connection, to support the complete modeling the variable structure system.
(2) The research fruit of our team is CVSDEVS (Yang et al. 2013), a DEVS-extended description norm for the complex variable structure system, which improves the ability of DEVS in describing the variability pattern and execution mode.

5.4.1.4 Time-Critical System Modeling Method

In the physical world, by contrast, processes are rarely procedural. CPS/IoT systems are composed by enormous physical units, which are highly time-critical and sensitive about the instruction sequences. The feedback loop between physical processes and computations encompasses sensors, actuators, physical dynamics, computation, software scheduling, and networks with contention and communication delays. To model those time-critical systems like CPS/IoT, research on the following key methods are required, including modeling of continuous and discrete hybrid system, the time-step and time-stamping methods, the back-roll methods for the synchronization error.

5.4.1.5 Deep Learning Modeling Method

With the rapid development of AI technology, an increasing number of CPS/IoT units are intelligent machines, which are featured with deep learning and self-adaptation. The complexities of system are rooted in the adaptability of system components that is good to model emergent properties. The typical modeling methods of complex self-adaptive systems are: Multi-Agent System modeling method, Cellular Automata modeling, and constrained generation process (CGP) model. Based on M2F and CGP description, we proposed constrained component model (CGP-EM, CGP-CM) (Li 2016).

5.4.1.6 High-Performance Parallel Algorithm

High-performance simulation algorithm for complex system is a kind of algorithm to employ high-performance simulation computers to solve complex system problem. In order to speedup simulation, our team focuses on research of three-level parallelization methods, including

(1) Task-level parallelization methods for large-scale problems: Research fruits include: Quantum multi-agent evolutionary algorithm (QMAEA) (Zhang et al. 2009); Cultural genetic algorithm (CGA) (Wu et al. 2012); Multi-group parallel differential evolution algorithm fusing azalea search (MPDEA);
(2) Federate-level parallelization methods between federates: Research fruits include: A federate-level parallelization method based on RTI (Li et al. 2012); An event-list based federate-level parallelization method (based on optimistic methods);
(3) Model/thread-level parallelization methods:. Research fruits include: Parallel algorithm of constant differential equations based on SMPS with load balance of right function; GA-BHTR: genetic algorithm based on transitive reduction and binary heap maintenance (Qiao et al. 2010).

5.4.2 SB-CPS/SB-IoT Simulation System Theory and Technology

5.4.2.1 CPS Oriented Multi-disciplinary Virtual Prototype Simulation System

Multi-disciplinary virtual prototype (MDVP) plays an increasingly important role in a wide range of engineering applications, especially the design, testing and evaluation of complex products, to integrate and optimize staff/organization, management and technology in the whole system and life cycle-related of SB-CPS/SB-IoT. The research focuses on:

(1) Heterogeneous computing environment for heterogeneous domain models, which would enable M&S of complex distributed systems which consists of physical system, sensor, controller, network and software (Li et al. 2008).
(2) Run time infrastructure (RTI) with improved time management service to address the challenge of inherent heterogeneity, distributed concurrency, and time criticality (Liu et al. 2014).
(3) Optimization framework of MDVP, which consists of the domain models, optimizer and the distributed interactive environment (Guo 2017).

5.4.2.2 CPS Oriented Complex System M&S Language

Complex system M&S language consists of the modeling environment, which supports model and experiment statements and syntax which describe the domain-dependent static and dynamic behavior, the translator/compiler, the libraries (for example, model libraries, algorithm libraries, function libraries), the simulation execution engine and the resulting processing software. The research focuses on

(1) Providing an easy, problem-oriented simulation modeling framework, which uses the similar descriptive statement of simulated problem (continuous, discrete, qualitative, etc.) with three parts including initialization part, model part and experiment part (Li et al. 2017b).
(2) Automatically generating programming languages, which would enable generation of M&S program suited to high-performance computer automatically (Li et al. 2017a).
(3) Making simulation more efficient to implement, making models easier to be accessed and modifiable, and automatically checking error. Configurable intelligent optimization algorithm was present (LaiLi et al. 2016 and Laili et al. 2011).

5.4.2.3 SB-CPS/SB-IoT High-Performance Simulation Computer System

There are two types of users: high-end users of high-performance SB-CPS/SB-IoT and massive number of end-users that acquire SB-CPS/SB-IoT cloud services on demand. These users perform three types of simulation (Digital, Man-in-the-loop, Hardware-in-the-loop/embedded simulation) with high-performance computer system that optimize the overall performance of modeling, simulation execution, and results analysis. The research focuses on

(1) High-performance simulation support hardware, which includes CPU + GPU-based heterogeneous high-performance computing systems, application-oriented high-bandwidth, low-latency interconnection network, high-capacity scalable global parallel I/O systems (Li 2016).
(2) Interface subsystem of hardware-in-the-loop/embedded simulation, which would enable simulation and decision support based on the massive, high-bandwidth, low-latency sensor data (Sato 2016).
(3) Parallel operating system technology to support three levels of parallelization. The algorithms are shown in Segment 4.1.6. An RTI on high-performance simulation computer was presented (Xing and Li 2016).
(4) High-performance visualization system, which includes simulation oriented hardware acceleration components based on GPU/CPU and GPU-based parallelization visualization system (Xu et al. 2014).

5.4.2.4 Cloud and Edge Merged Smart Cloud Simulation System

Smart cloud simulation, which adopts and extends the concept of cloud computing, virtualizes various simulation resources and capabilities (SR/Cs) and builds an SR/C pool (cloud) to deliver on-demand simulation services for the whole life cycle activities of SB-CPS/SB-IoT, through network (including internet, IoT, telecommunication network, broadcasting network, mobile network, etc.) anytime and anywhere. The research focuses on

(1) Unified virtualization and management technology for SR/Cs on cloud and edge, which would enable building ubiquitous networks based, "human, machines, things, environments, information" integrated Internet of SR/C service (Lin et al. 2013).
(2) Operation environment construction technology for SR/Cs on cloud and edge, which would address the challenges of merging cloud simulation and edge simulation. The general idea and process were proposed in ASIASIM2011 (Li et al. 2012).
(3) Collaborative and interoperability technology for SR/Cs on cloud and edge, which involves object management, connection management, time management, load balance management, RTI agent. A layered parallel and discrete simulator oriented to multi-core clusters (Ivy-DS) was presented in (Yang et al. 2016).

5.4.2.5 CPS Oriented Embedded Simulation System

Embedded simulation system plays an increasingly important role in CPS, which helps the physical system to simulate the more complicated case online and aids the decision-making based on the status of the system. The research focuses on

(1) Pervasive simulation to perceive the status of the physical system and the requirement of the user smartly, which was presented in (Zhen et al. 2007).
(2) Edge simulation, which would enable digital twins based on prescription to respond the change of the status in time (Shi and Dustdar 2016).
(3) High-performance simulation to simulate the thousands of the case for decision-making (Guo 2017), which would address the real-time decision based on the simulation.

5.4.3 SB-CPS/SB-IoT Simulation Application Engineering Theory and Technology

5.4.3.1 SB-CPS/SB-IoT VV&A Technology

Verification, Validation and Accreditation (VV&A) technology should be applied in the whole life cycle activities of simulation system. Validation could ensure that the system models express the user's research intentions; verification could ensure that simulator processes a satisfactory range of accuracy; accreditation could ensure that the simulation results satisfy user's specified criteria. The research focuses on:

(1) VV&A of the whole life cycle, the whole system, the whole staff and the full range of management to ensure the consistency and high credibility of SB-CPS/SB-IoT (Qian et al. 2016; Fang et al. 2005).
(2) Providing knowledge management, data sharing, and integrated platform with distributed VV&A tools (Jiao et al. 2012) for SB-CPS/SB-IoT, which would address the challenges of valuation of complex SB-CPS/SB-IoT system.

5.4.3.2 SB-CPS/SB-IoT Simulation Result Processing Technology

Simulation result processing technology enables the simulation experiment data collection, visualized processing and analysis, and intelligent evaluation for effective reuse and value-added use of simulation results in SB-CPS/SB-IoT. The research focuses on:

(1) Online, dynamic simulation results analysis and processing technology, which would enable efficiently processing amount of simulation result data for SB-CPS/SB-IoT (Suzumura 2014).
(2) Open, customized, reusable intelligent analysis model (Li et al. 2011a, b), which would address the data processing for the full life cycle of all kinds of simulation (Digital, Man-in-the-loop, Hardware-in-the-loop/embedded simulation).

5.4.3.3 VR/AR-Based Visualization System

VR/AR-based Visualization system could support the simulation results management, analysis, and evaluation, which could effectively enhance the understanding and application of the result data. The research focuses on

(1) Presence and interaction in mixed reality environments (Egges et al. 2007), which would address the challenges of using VR/AR in the scene of CPS and IoT seamlessly.
(2) VR/AR-based Reconstruction of product application environment (Carozza et al. 2013), which would enable innovative simulation application for SB-CPS/SB-IoT.

5.5 Some Application Cases, E.G. Simulation of Smart Cities, Smart Manufacturing, Are Illustrated

SB-CPS/IoT systems can be divided into three different levels in practice: device/equipment level, system level, and SoS level. Industrial robot is a typical CPS system in the device/equipment level. In practice, application of industrial robot is inseparable of the support of simulation technology. In this section, we present three applications of simulation-based CPS/IoT systems in intelligent manufacturing and intelligent traffic domains. Cyber-factory and cloud manufacturing corresponding to system level and SoS level SB-CPS/IoT systems, and intelligent traffic is a system level SB-CPS/IoT application.

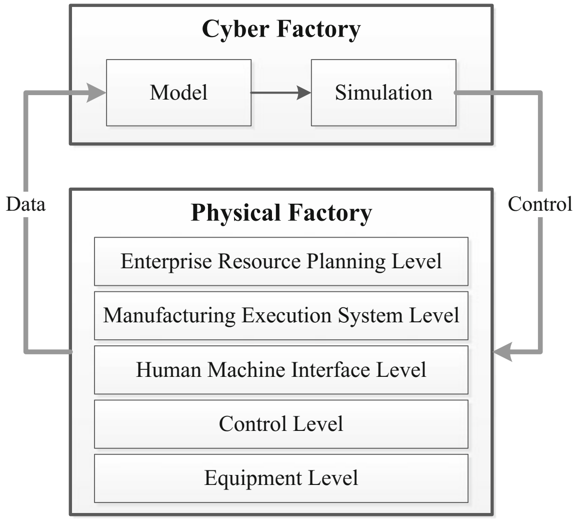
5.5.1 Intelligent Manufacturing

(1) Cyber-Factory

Success in cyber-physical systems engineering strongly depends on proper application of model-based systems engineering (Feeney et al. 2017). The assembly of CPS, IoT, Big Data technologies has led to the new concept, "Smart Factory", in the field of manufacturing and the entire value chain from product design to delivery is digitalized and integrated.

CPS refers to the convergence of the physical and computing (cyber) systems over the network. When applied to production, CPS is specialized in Cyber-Physical Production Systems (CPPS). CPPS consists of autonomous and cooperative elements and subsystems that connect with each other in situation dependent

Fig. 5.6 CPPS architecture



ways, on and across all levels of production. Figure 5.6 illustrates the architecture of CPPS (Jeon et al. 2016).

CPPS architecture refers to that the cyber and the physical factories are synchronized in time by exchanging data and control messages between them. The simulation in the cyber factory takes data in real-time from the physical factory so that the accurate model (the physical factory model) can be defined. Once the simulation is done, the simulator generates control messages and sends back to the actuators in the physical factory so that the flexible adaptation of the production chain and the production optimization are possible (Jaeho 2016).

(2) Cloud Manufacturing system

CASICloud (http://www.casicloud.com/) is China's first independently developed Internet-based public cloud service system for manufacturing industry. CASICloud aims at integrating Internet technologies with intelligent manufacturing and facilitating resources sharing. It targets industrial enterprises both in China and overseas. CASICloud focus on helping users with crowd sourcing and industrial products trade, as well as starting their own business.

Since December 2015, CASICloud has been online for over a year, attracting more than 230,000 registrations and releasing over 43 billion CNY CASIC business demands regarding all aspects of the manufacturing industry. Over 1000 innovation and entrepreneurship projects have been released online, and cooperation with international intelligent manufacturing and scientific services is underway (Li et al. 2017a).

By virtualizing the manufacturing resources and manufacturing capabilities in the physical layer, CASICloud can provide services in the whole life cycle of

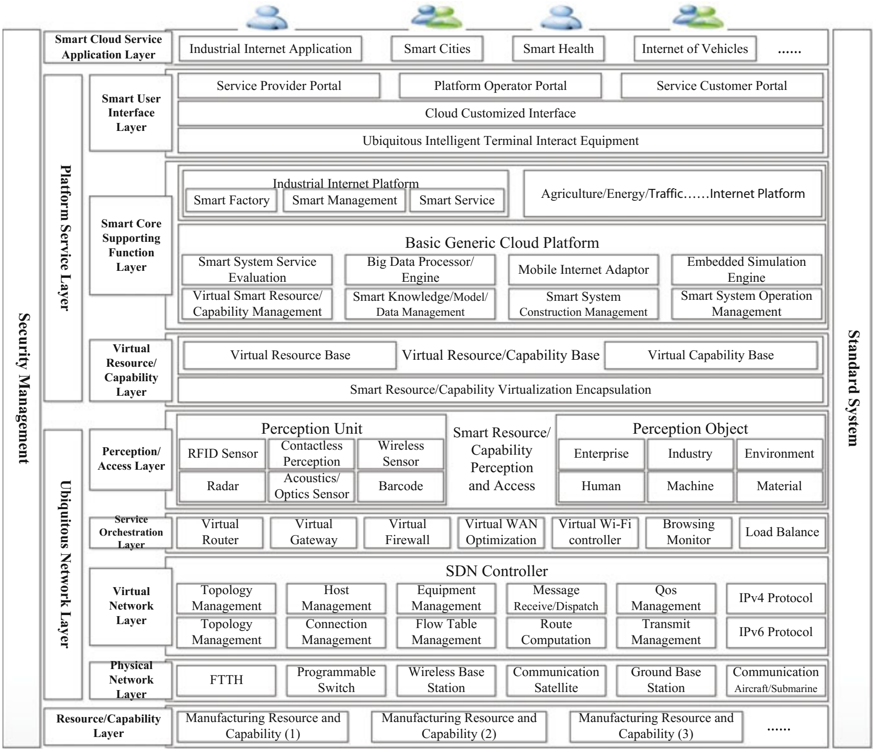


Fig. 5.7 System architecture of CASICloud

manufacturing on demand. Modeling and simulation technology plays a critical role in the operation of CASICloud. CASICloud is a typical SB-CPS/IoT system at SoS level. The system architecture of CASICloud is presented in Fig. 5.7.

5.5.2 *Simulation of Intelligent Traffic*

Smart Cities are cities enhanced with a technological infrastructure that enables a more intelligent use and management of its resources. They are currently seen as a powerful way of improving the quality of life of its citizens (Santana et al. 2016). Making cities smarter can help optimize resource and infrastructure utilization in a more sustainable way.

Intelligent traffic is an important aspect of smart cities. In recent years, popularity of private cars is getting urban traffic more and more crowded. As a result, traffic is becoming one of the important problems in big cities all over the world. SB-IoT provides a new solution for this problem.

The IoT is based on the Internet, network wireless sensing, and detection technologies to realize the intelligent recognition on the tagged traffic object,

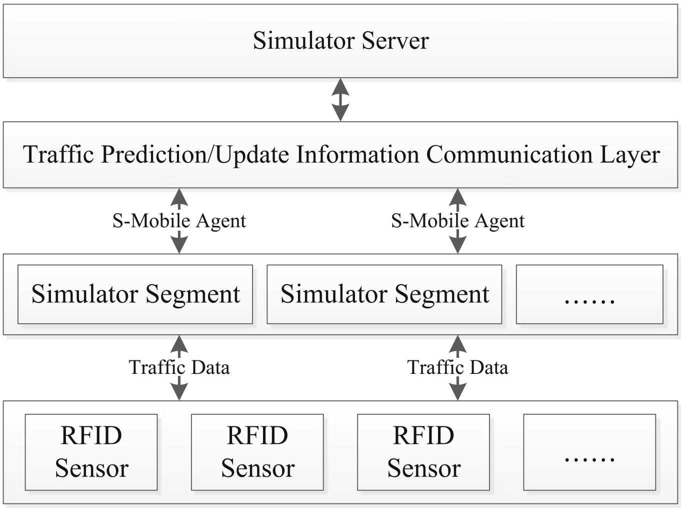


Fig. 5.8 General model of distributed traffic simulation framework

tracking, monitoring, managing, and processed automatically. The simulation-based approach introduces the use of an active radio-frequency identification (RFID), wireless sensor technologies, object ad hoc networking, and Internet-based information systems in which tagged traffic objects can be automatically represented, tracked, and queried over a network. The general model of distributed traffic simulation framework described in Fig. 5.8 (Al-Sakran 2015). The simulation server disseminates information among the simulator segments, coordinates all simulators' segments and provides a predictive model of traffic conditions in specified traffic areas by analyzing and integrating the results of distributed simulators of those areas. The simulation server maintains state information of current and future operations of the traffic network such as flow rates, average speed, and the time when that information was generated (Al-Sakran 2015).

5.6 Conclusion

5.6.1 The Impact of SB-CPS/SB-IoT on Society and Economy, People's Livelihood, National Security

5.6.1.1 The Era of "Internet+ Artificial Intelligent+" Is Coming

(1) The network in the era of "Internet+ Artificial Intelligent+" refers to the pervasive network including Internet, IoT, Mobile Internet, the Satellite Network, Space-Ground Integrated Network, Next Generation Internet, etc.

(2) The key technology in the era of “Internet + Artificial Intelligence” is deeply integrated with seven types of new technologies shown as follows:

(a) New Internet Technology (Internet, IoT, Mobile Internet, the Satellite Network, Space-Ground Integrated Network, Next Generation Internet, etc.);
(b) New Information Technology (Cloud Computing, Big Data, High-performance Computing, M&S, etc.);
(c) New Artificial Intelligent Technology (Big Data based A.I., Group Intelligence, Human-machine Intelligence, etc.);
(d) New Power Technology (Solar Power, Wind Power, Bio-energy, Geothermal Energy, Ocean Energy, Chemical Energy, Nuclear Energy, etc.);
(e) New Material Technology (Metallic Materials, Inorganic Non-metallic Materials, Organic Polymer Materials, Advanced Composite Materials, etc.);
(f) New Biological Technology (New Biological Medicine, Biological Technology, Green Manufacturing Advanced Biomedical Materials, Advanced Biotechnology, Common Use of Biological Resources, Biological Safety, Life Science Instruments, Synthetic Biology, Biological Data, Regenerative Medicine, 3D Bio Printing, etc.);
(g) New Application Domain Technology (Economy, National Security, Society, etc.).

(3) The ecosystem of the era of “Internet + Artificial Intelligence” refers to: “Pervasive Connection, Data-driven, Sharing and Serving, Cross-border Fusion, Self-intelligence, Mass Innovation”, which depict the ecosystem of “Internet + Artificial Intelligence”.

5.6.1.2 Simulation-Based CPS/IoT Is Promoting the Changes in the New Paradigms, New Methods and New Ecosystems in All CPS/IoT Domains

Modeling and simulation technology has involved through the whole life cycle activities of CPS and IoT systems, including argumentation, design, experiment, production, operation, management and service. It is promoting the changes in the new paradigms, new methods and new ecosystems in all CPS/IoT domains.

For example

(1) New paradigm, new method, and new ecosystem in intelligent manufacturing

(a) New paradigm in intelligent manufacturing: user centered, inter-connected, service-oriented, collaborative, individualized (customized), flexible and social intelligent manufacturing and serving.

(b) New method in intelligent manufacturing: "human, machines, things" integrated system characterized by "digitalization, things networking, virtualization, service oriented, collaboration, customization, networking, and intelligence".
(c) New ecosystem in intelligent manufacturing: characterized by "pervasive connection, data-driven, sharing and serving, cross-border fusion, self-intelligence and mass innovation".

(2) New paradigm, new method, and new ecosystem in smart city

(a) New paradigm in smart city: user centered, "human, machines, things" integrated, inter-connected, service-oriented, collaborative, individualized, flexible and social smart city operation.
(b) New method in smart city: Based on the ubiquitous network, using digital and network, intelligent methods integrated by four technologies (advanced information and communication technology, intelligent technology, system engineering technology and city operation and management technology), building a SR/Cs' cloud (network) of smart city for the citizen, the government and the enterprises with the synchronous development of the industrialization, informatization, urbanization, agricultural modernization.
(c) New ecosystem in smart city: characterized by "pervasive connection, data-driven, sharing and serving, cross-border fusion, self-intelligence and mass innovation".

5.6.2 Further Works on SB-CPS/IoT

CPS/IoT is a basic component of various intelligent application systems in the age of "Internet+ Artificial Intelligence+". CPS/IoT are typical complex systems in which human, machine, material, environment and information sectors are intelligently connected in depth. SB-CPS/IoT related technology, industry, and application should be further developed coordinately to adapt to the era of "Internet + Artificial Intelligent+":

(1) For development of technology, more attention should be paid to the integration of simulation science and technology, advanced information and communication science and technology, advanced intelligent science and technology as well as application technology. Enhance research on heterogeneous integration techniques in SB-CPS/SB-IoT system techniques as well as big data techniques, high-performance simulation/computing techniques, advanced artificial intelligence techniques and AR/VR techniques in SB-CPS/SB-IoT platform techniques. Enhance research on new model, new procedure, new means and new format of modeling and simulation in

SB-CPS/SB-IoT systems. Enhance research on new business model that in line with sharing economy model. Pay attention to research on security technology as well as related standards and evaluation index system. Pay attention to continuous construction of hierarchical technical innovation system.

(2) For development of industry, enhance the SB-CPS/SB-IoT system toolkit and platform development industry. Enhance the system construction and operation industry in SB-CPS/SB-IoT industrial chain.

(3) For development of applications, promote application demonstration of SB-CPS/SB-IoT system applications. The application and implementation of SB-CPS/SB-IoT systems are complex system engineering, so emphasis should be made on (a) domain-oriented, (b) revolutionary new paradigm, new method and new ecosystem, (c) the integration, optimization and intelligence of system, (d) the strategic development plan and (e) making phased implementation schemes properly.

Review questions

1. What is the difference between CPS and IoT?.
2. How can you use the modern M&S technologies, such as complex hybrid network system modeling, quantitative and qualitative hybrid system modeling, big data-based modeling, Multi-disciplinary virtual prototype simulation system, high-performance simulation computer and cloud simulation, to improve the productivity, quality and cost of the whole life cycle activities of CPS and IoT systems, including argumentation, design, experiment, and evaluation?
3. How can you use the modern M&S technologies such as high-performance simulation, cloud, and edge merged simulation as embedded simulation, a basic component, to support the operation activity of the CPS/IoT system in unit level, system level and system-of-system level?

References

Al-Sakran, H. O. (2015). Intelligent traffic information system based on integration of Internet of Things and Agent technology. International Journal of Advanced Computer Science and Applications (IJACSA), 6(2), 37–43.

Carozza, L., Bosché, F., & Abdel-Wahab, M. (2013). Image-based localization for an indoor VR/AR construction training system. CONVR 2013.

Chinese CPS Development Forum (2017). White Paper of Cyber-Physical System.

Egges, A., Papagiannakis, G., & Magnenat-Thalmann, N. (2007). Presence and interaction in mixed reality environments. The Visual Computer, 23(5), 317–333.

Fan, S., Li, B. H., Chai, X., Hou, B., Wang, W., & Wu, Y. (2009, July). Research on complex system quantitative and qualitative synthetic M&S platform. In Proceedings of the 2009 Summer Computer Simulation Conference (pp. 208–215). Society for Modeling & Simulation International.

Fang, K., Yang, M., Wang, Z. C. (2005). A VV&A evaluation system based on hierarchical evaluation. Journal of Harbin Institute of Technology, 12(2), 184–188.

Feeney, A. B., Frechette, S., & Srinivasan, V. (2017). Cyber-Physical Systems Engineering for Manufacturing. In Industrial Internet of Things (pp. 81–110). Springer International Publishing.

Guo L. Q. (2017). A Cloud Simulation Based Environment for Multi-disciplinary Collaborative Simulation and Optimization. In Proceedings of The 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design. IEEE. (Accepted)

International Telecommunications Union (2005). ITU Internet Reports 2005: Internet-of-Things.

Telecommunication Standardization Sector of ITU (2012). Overview of the Internet of things.

Jeon, J., Kang, S., Chun, I. (2016). CPS-based Model-Driven Approach to Smart Manufacturing Systems. In Proceedings of the 5th International Conference on Intelligent Systems and Applications (pp. 133–135). IARIA.

Jiao, S., Li, W., Ma, P., & Yang, M. (2012, July). Knowledge-Based Simulation Experiment Data Integrative Analysis Technology. In Principles of Advanced and Distributed Simulation (PADS), 2012 ACM/IEEE/SCS 26th Workshop on (pp. 165–167). IEEE.

LaiLi, Y., Zhang, L., & Tao, F. (2011). Energy Adaptive Immune Genetic Algorithm for Collaborative Design Task Scheduling in Cloud Manufacturing System. In Proceeding of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM2011)

LaiLi, Y., Zhang, L., & Tao, F. (2016). Rotated Neighbor Learning-based Auto-Configured Evolutionary Algorithm. Science China Information Sciences, online published.

Li, B. H. (2016). Study, Practice and Thinking on High Performance Modeling & Simulation Technology of Complex System. Keynote speech on the joint conference of the 16th Asia Simulation Conference and the 2016 Autumn Simulation Multi-Conference.

Li, B. H., Chai, X., Zhang, L., Hou, B., Lin, T. Y., Yang, C., & Li, T. (2012). New advances of the research on cloud simulation. In Advanced methods, techniques, and applications in modeling and simulation (pp. 144–163). Springer Japan.

Li, B. H., Hou, B. C., Yu, W. T., Lu, X. B., & Yang, C. W. (2017a). Applications of artificial intelligence in intelligent manufacturing: a review. Front. Inform. Technol. Electron. Eng, 18 (1), 86–96.

Li, B. H., Song, X., Zhang, L., Liu, J., Chi, P., Lin, T., & Li, T. (2017b) CoSMSOL: Complex system modeling, simulation and optimization language. International Journal of Modeling, Simulation, and Scientific Computing, 1741002.

Li, B. H., Zhang, Y., Tan, L., Zhang, Z., Chai, X., Hou, B., & Du, Z. (2008, October). Research and Application on Virtualization Service Oriented Infrastructure for Networkitized M&S. In Grid and Cooperative Computing, 2008. GCC'08. Seventh International Conference on (pp. 658–664). IEEE.

Li, T., Li, B. H., Chai, X. D., & Fan, S. (2011a). Research on knowledge modeling and joint simulation method of complex qualitative system. Journal of System Simulation, 23(6), 1256–1260.

Li, T., Li, B. H., Chai, X. D., & Yan, X. F. (2011b). Meta modeling framework for complex product multidiscipline virtual prototyping. Computer integrated manufacturing systems, 17(6), 1178–1186.

Lin, T. Y., Li, B. H., & Yang, C. (2013, September). A Multi-centric Model of Resource and Capability Management in Cloud Simulation. In Modelling and Simulation (EUROSIM), 2013 8th EUROSIM Congress on (pp. 555–560). IEEE.

Liu, S. T., Dai, S. L., & Zhang, Y. L. (2014). Improvement of HLA/RTI real-time performance by scheduling periodic and aperiodic tasks. Journal of Beijing University of Aeronautics & Astronautics, 40, 110–114.

Liu, T., Chen, Z., & Chen, X. R. (2005). A Brief Review of Complex Networks and Its Application [J]. Systems Engineering, 6, 000.

Pu, C. (2011, July). A world of opportunities: CPS, IoT, and beyond. In Proceedings of the 5th ACM international conference on Distributed event-based system (pp. 229–230). ACM.

Qian, X., Li, W., & Yang, M. (2016). Two-stage nested optimization-based uncertainty propagation method for model calibration. International Journal of Modeling, Simulation, and Scientific Computing, 7(01), 1541003.

Qiao, K., Tao, F., Zhang, L., & Li, Z. (2010, October). A GA maintained by binary heap and transitive reduction for addressing PSP. In Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on (pp. 12–15). IEEE.

Santana, E. F. Z., Bastista, D. M., Kon, F., & Milojicic, D. S. (2016). SCSimulator: An Open Source, Scalable Smart City Simulator. In Tools Session of the 34th Brazilian Symposium on Computer Networks (SBRC). Salvador, Brazil.

Sato, T. (2016). Paradigm change for new era IoT society. Keynote speech on the joint conference of the 16th Asia Simulation Conference and the 2016 Autumn Simulation Multi-Conference.

Shi, W., & Dustdar, S. (2016). The promise of edge computing. Computer, 49(5), 78–81.

Suzumura, T. (2014, March). Extreme big data processing in large-scale graph analytics and billion-scale social simulation. In Proceedings of the 5th ACM/SPEC international conference on Performance engineering (pp. 1–2). ACM.

Wu, Y., Zhang, L., Tao, F., Shen, Y., Liu, D., & Mu, L. (2012). Modeling and High Performance Computing Analysis of Three-Dimensional Electromagnetic Environment. In System Simulation and Scientific Computing (pp. 25–33). Springer Berlin Heidelberg.

Xing, C., Li, B. H. (2016). Research on rti communication mechanism on infiniband network architecture. Acta Electronica Sinica.

Xu, Y., Tan, G., Li, X., & Song, X. (2014, May). Mesoscopic traffic simulation on CPU/GPU. In Proceedings of the 2nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (pp. 39–50). ACM.

Yang, C., Chi, P., Song, X., Lin, T. Y., Li, B. H., & Chai, X. (2016). An efficient approach to collaborative simulation of variable structure systems on multi-core machines. Cluster Computing, 19(1), 29–46.

Yang, C., Li, B. H., Chai, X., Chi, P., & Ren, L. (2013, September). An efficient dynamic load balancing method for simulation of variable structure systems. In Modelling and Simulation (EUROSIM), 2013 8th EUROSIM Congress on (pp. 525–531). IEEE.

Zhang, L., Wu, Y., & Huntsinger, R. C. (2009, July). Key issues in electromagnetic environment modeling & simulation. In Proceedings of the 2009 Summer Computer Simulation Conference (pp. 88–94). Society for Modeling & Simulation International.

Zhen, T., Li, B. H., Xu-Dong, C., & Chang-Feng, S. (2007, October). Elementary Studies on Pervasive Simulation Grid. In Asian Simulation Conference (pp. 276–286). Springer Berlin Heidelberg.

Author Biographies

Bo Hu Li is an Academician of Chinese Academy of Engineering, a Ph.D. adviser and the Honorarium President in the School of Automatic Science and Electrical Engineering in Beijing University of Aeronautics and Astronautics (BUAA). He is also the Co-President of China Simulation Federation (CSF) and the Co-Chief-Editor of the international journal "International Journal of Modeling, Simulation, and Scientific Computing" (IJMSSC). He was the Director of Beijing Institute of Computer Application and Simulation Technology and Beijing Simulation Center, as well as the President of the School of Automatic Science and Electrical Engineering in BUAA. He was the first President of Federation of Asian Simulation Societies (ASIASIM). In addition, He was the Director of Expert Committee of Contemporary Integrated Manufacturing System (CIMS) Subject in Chinese National High Technology Research and Development Plan (863 Plan). He authored or co-authored 300 papers, 13 books and 4 translated books in the fields of modeling and simulation, concurrent engineering, cloud manufacturing and manufacturing informatization. He has got one first class and three second class scientific Science and Technology Progress Award from China State. In 2012, He got the SCS Life time Achievement Award (SCS Modeling and Simulation Hall of Fame) of the Society for Modeling and Simulation International (SCS).

Lin Zhang is a full professor of Beihang University. He received the B.S. degree in 1986 from the Department of Computer and System Science at Nankai University, China. He received the M.S. degree and the Ph.D. degree in 1989 and 1992 from the Department of Automation at Tsinghua University, China. From 2002 to 2005 he worked at the US Naval Postgraduate School as a senior research associate of the US National Research Council. His research interests include service-oriented modeling and simulation, agent-based control and simulation, cloud manufacturing, model engineering for simulation. Currently, he served as the Past President of the Society for Modeling & Simulation International (SCS), a Fellow of the Federation of Asian Simulation Societies (ASIASIM), the executive vice president of China Simulation Federation (CSF), an IEEE senior member, a chief scientist of 863 key projects, and associate Editor-in-Chief and associate editors of six peer-reviewed journals. He authored and co-authored 200 papers, five books and chapters. He received the National Award for Excellent Science and Technology Books in 1999, the Outstanding Individual Award of China High-Tech R&D Program in 2001, the 2nd prize of Ministry of Education Science and Technology Progress Award in 2013, the National Excellent Scientific and Technological Workers Awards in 2014.

Tan Li is the vice general manager of ASM (Aerospace Smart Manufacturing Co. Ltd.), which is a leading company in cloud manufacturing and industrial internet. In 2011, he received the Ph.D. degree from School of Automation Science and Electrical Engineering in Beihang University, China. He was the co-proposer of the concept of “Cloud Simulation” and “Cloud Manufacturing” with his supervisor, Professor Bo Hu Li. He and his team developed the outstanding collaborative simulation platform COSIM, the first complex system simulation language CSMSL, and the leading CPS/IoT platform CASICloud in China. And he has published over 30 articles and two books in cloud simulation, high-performance simulation language and cloud manufacturing. He is also member of the China National Standardization Technical Committee in Automation and Manufacturing.

Ting Yu Lin is a senior engineer in Beijing Complex Product Advanced Manufacturing Engineering Research Center of Beijing Simulation Center and State Key Laboratory of Intelligent Manufacturing System Technology of Beijing Institute of Electronic System Engineering. In 2007 and 2014, he received the B.S. degree and the Ph.D. degree from School of Automation Science and Electrical Engineering in Beihang University, China. He authored and co-authored 40 papers, three books and chapters in the fields of complex system modeling and simulation and cloud manufacturing. He is now a member of China Simulation Federation (CSF).

Jin Cui is currently a Ph.D. student at School of Automation Science and Electrical Engineering in Beihang University, China. His research interests include cloud manufacturing, quantitative evaluation techniques and data analysis techniques. He is a student member of the Society for Modeling & Simulation International (SCS), a member of China Simulation Federation (CSF). Now he serves as the vice president of SCS Beihang Student Chapter (SCS-BH-SC).

Chapter 6
Simulation-Based Complex Adaptive Systems

Saurabh Mittal and José L. Risco-Martín

Abstract Complex Adaptive Systems (CAS) are systems that display two primary characteristics: emergent behavior, and adaptive behavior. Emergent behavior manifests in a system comprising of large number of components, often considered as agents, engaged in multi-level interactions. Adaptive behavior manifests at the agent–environment boundary when the agent situates itself in an environment. Modeling a CAS has been a challenge due to limitations in bringing these two aspects together in a single formal specification in a computational environment. Lack of simulation environment for a CAS model adds further problem in validation and verification of a CAS model. Computationally, the emergent behavior can be understood using today's latest technology of feature engineering, Deep Learning, and data analytics using Big Data. This would facilitate the identification of various holistic behaviors and their classification that would aid designing various observers for detecting the emergent behavior in a computational environment. This aspect is largely bottom-up. Once various observers are computationally available, they can be integrated in the agent behavior repertoire so that the emergent behavior, that is now detectable and perceivable at the agent-environment boundary, can be used and acted-upon by the agent. This situates the agent in the environment and manifests as adaptable behavior. This activity is top-down as there is a conscious design process (done by a human) that is employed for such behavior refinement. This chapter will discuss the state of the art in computational and simulation support needed and provides foundation to manifest accurate emergent behavior in a computational environment as a means to perform CAS engineering.

The authors affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved for public release: Case: 17-1519.

S. Mittal (✉)
The MITRE Corporation, McLean, USA
e-mail: smittal@mitre.org

J.L. Risco-Martín
Complutense University of Madrid, Madrid, Spain
e-mail: jlrisco@ucm.es

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_6

Keywords Complexity · Emergent behavior · Complex adaptive systems · System of systems · Co-simulation · Multi-paradigm modeling · Cyber-physical systems · Cyber complex adaptive systems · Network science · Adaptive · Multi-agent systems · Evolutionary computing · Emergence complexity cone · Hyper-heuristics

6.1 Introduction

System complexity continues to grow by leaps and bounds. Multi-level complexity is a fundamental nature of heterogeneous systems today. To describe this new class of super complex systems in a man-made world, labels such as system of systems (SoS), cyber-physical systems (CPS) (Lee 2008), complex adaptive systems (CAS) (Holland 1992), and cyber CAS (CyCAS) (Mittal 2013b, 2014) are used interchangeably. All of them are multi-agent systems, i.e., have large number of agents, are contextualized in the environment and manifest emergent behavior. The constituting agents are goal-oriented with incomplete information at any given moment and interact among themselves and with the environment. SoS is characterized by the constituent systems under independent operational and managerial control. CAS is an SoS where constituent systems can be construed as agents that interact and adapt to the dynamic environment. Cyber CAS is a CAS that exists in a netcentric environment (for example, Internet) that incorporates human elements where distributed communication between the systems and various elements is facilitated by agreed upon standards and protocols. CPS is an SoS wherein the constituent physical and embedded systems are remotely controlled through the constituent cyber components. In modern times, the Internet of Things (IoT) is beginning to incorporate these characteristics and is becoming a significant contributor to the increase in complexity. However, the IoT phenomenon is still in the formative stages of an apparently exponential growth. Designing these systems is equally complex and methodologies available through traditional systems engineering practices fall short of engineering these super complex systems.

CAS is not a new concept. Fifty years ago, in his book, Alexander (1964) drew the following context:

- "Today more and more design problems are reaching insoluble levels of complexity"
- "At the same time that problems increase in quantity, complexity and difficulty, they also change faster than before"
- "Trial-and-error design is an admirable method. But it is just real world trial and error which we are trying to replace by a symbolic method. Because trial and error is too expensive and too slow."

This scenario continues to be applicable even today, and probably will be 50 years from now as well. The sheer lack of methodologies to do CAS engineering in a heterogeneous environment makes it a hard problem. Validation and verification for CAS engineering is being defined (Mittal and Zeigler 2017). The use of real-world data and of artificial data to test a hypothetical CAS is still being characterized. Not only that, the properties of CAS do not belong to a single domain of interest but are cross-domain and multidisciplinary. Approaches that leverage data analytics, speed iterations, built in agility, and ensure a holistic view for robust decision-making that are essential to solve complex engineering programs are in the process of being incorporated in the CAS engineering process. Consequently, engineering CAS elements as well the processes to facilitate CAS engineering are being worked out. Due to the complexity of such systems, these systems exhibit emergent behavior. Efforts are underway to harness emergent behavior as an essential component of the CAS engineering process (Mittal and Rainey 2015). To address the growing need for studying CAS, and develop engineering methodologies for such multidisciplinary systems, various scientific journals have recently been launched:

- Journal of Cognitive Systems Research, Elsevier
- Information Sciences, Elsevier
- Complex Adaptive Systems Modeling, Springer
- Complexity, Hindawi, Wiley & Sons.

Traditionally, modeling and simulation has supported the systems engineering process across all its phases. Paradigms like model-based system engineering (MBSE), model-based development (MBD), and model-based engineering (MBE) are often used to describe the usage of modeling practices in systems and software engineering. Many times, modeling is performed without the simulation. In that role, modeling acts as a validation mechanism to bring a common understanding to all the stakeholders that include the end-users (who use the system), the engineers (who make the system), and the business holders (who invest in the system). These validated models serve as architectural blueprint. Such models are usually static in nature and the verification or correctness of these models rests with the modeling workbench used to create these models. The metamodels implemented in the modeling editors are the foundation on which the modeling representation rests. To evaluate the dynamic behavior of these models, simulation is warranted.

With CAS, as both the model and the data it is dealing with, are multidisciplinary, the CAS model is constantly evolving. The traditional use of M&S in a multidisciplinary environment is a challenge and needs to be explored in a cross-disciplinary manner (Mittal and Zeigler 2017). Some of the research centers are specifically focused on the very subject for a holistic cross-disciplinary undertaking:

- Center for Comp. Analysis of Social and Org. Systems, Carnegie Mellon University
- Center for Complex Systems Research, University of Illinois at Urbana-Champaign
- Center for the Study of Complex Systems, University of Michigan

- Evolution, Complexity, and Cognition group, Vrije Universiteit Brussel
- Institute for Complex Engineered Systems, Carnegie Mellon University
- New England Complex Systems Institute
- Northwestern Institute on Complex Systems, Northwestern University
- Santa Fe Institute, New Mexico.

Modeling is an integral part of studying complex systems. However, without a failsafe validation process, the model can fall out-of-sync quickly rendering the model useless. Much of the model validation depends on the data validation. Data validation depends on the data exchanged between the actual system and its environment, in comparison to the data exchanged between the model and the environment. Acquisition of structured data from natural systems and man-made systems that can capitalize on various data-driven analyses is a growing challenge. Data-driven methodologies need to scale up in a multidisciplinary environment as well. If the modeling workbenches can support agile modeling processes, the simulation infrastructure needed to validate the dynamical behavior should be made agile as well.

Emergent behavior, a macro behavior that is irreducible at the micro-level, is a characteristic property of any CAS, along with many other properties like self-organization, nonlinearity, and order/chaos dynamics (Mittal 2013a). Reproducing emergent behavior in an artificial environment, such as through M&S endeavor is a nontrivial problem. There are two fundamental reasons. By the definition of emergent behavior by Ashby (1956), the source of emergent behavior of a system results from an incomplete understanding of the system. So, to model a CAS, do we have complete understanding of CAS, to begin with? The answer is clearly no because, emergent behavior is a by-product of all the structural and behavioral richness in a natural CAS. This emergent behavior results in a dynamic environment and dynamic agent behavior that adapts to the dynamic environment and eventually leads to agent learning and adaptation as it survives in that environment. In a modeling endeavor, much of the low-level detail is deliberately ignored to manage the model's complexity. Abstraction results in loss of information at a finer level. Regardless, a model can be made to produce the same emergent behavior with the support of a hypothesis or theoretical constructs. To check the hypothesis, the model must be simulated on a time-base to ensure the correct dynamical behavior. This leads us to the second reason. Developing a multi-agent co-simulation platform in a multidisciplinary environment is also a nontrivial challenge and out-of-order events and computational complexity of the simulator execution introduce unintended behaviors that contribute to inaccurate emergent behavior. Validation and verification of agent-based modeling environments is an active area of research (Yilmaz 2006; Arifin and Madey 2015).

This chapter discusses the complexity inherent in CAS modeling, the multi-faceted data-driven methodology and the supporting simulation infrastructure using a co-simulation methodology. Having a robust co-simulation infrastructure is a must-have to eliminate unintended emergent behaviors arising out of computational simulation environment (Mittal and Zeigler 2017). Only ensuring that will keep the focus of CAS engineers toward modeling the desired and accurate emergent behaviors.

The chapter is structured as follows. Section 6.2 discusses the complexity in multidisciplinary CAS engineering. Section 6.3 discusses the adaptive aspect in CAS engineering. The nature of emergent behavior, its taxonomy and how it is reproduced in a modeling and simulation environment is discussed in Sect. 6.4. The data aspect supporting model evolution is discussed in Sect. 6.5. The co-simulation architecture for simulation-based CAS engineering is described in Sect. 6.6, followed by conclusions and future work in Sect. 6.7.

6.2 Complexity in CAS Structure Modeling

A basic definition of CAS was provided by Holland (1992):

> CAS are systems that have a large number of components, often called agents that interact and adapt or learn

More specific definitions appreciating the associated complexity started appearing after a decade later. The earliest and the most cited definition we could find is provided by Plesk and Greenhalgh (2001):

> A complex adaptive system is a collection of individual agents with freedom to act in ways that are not always totally predictable, and whose actions are interconnected so that one agent's actions change the context for other agents.

Miller and Page (2009) discussed the subject of CAS and the difference between a complicated and complex system, and how computational models can help describe a complicated system but hit a boundary when it comes to complex systems with adaptive behavior. While they describe the CAS components in an illustrative and computational manner, they did not actually provide a definition of CAS. A precise definition of an emerging new concept is an important undertaking, as it will define what it "is" and "is not." This also hits at the heart of how M&S can support the concept definition activity and thereby, advance science. Indeed, if a concept is expressed precisely, it will be reflected accordingly in the created model, and can be validated. Failure to do so will introduce ambiguity and uncertainty.

System modeling begins with understanding the structure and behavior of the system. Once sufficient understanding is available, model-based design and analyses can proceed. To understand the structure and behavior of CAS, thereby to perform MBSE for CAS, Mittal (2013b) stated:

> A CAS is a complex, scale-free collectivity of interacting adaptive agents, characterized by high degree of adaptive capacity, giving them resilience in the face of perturbation. Indeed, designing an artificial CAS requires formal attention to these specific features. Complexity is a phenomenon that is multivariable and multi-dimensional in a space-time continuum.

From the structural perspective, a multi-agent system is analogous to an SoS. The connectedness of each of these systems is well researched in the works of

Table 6.1 Scale-free characteristics of complex adaptive systems, summarized from Mittal (2013a)

ID	Feature	Description
1	Incremental growth	Incremental linking to other agents/systems in a persistent environment (an environment that has a spaciotemporal nature and preserves history)
2	Self-organization	Agents/systems may organize themselves into clusters, groups for a common objective
3	Critical state-transition	At appropriate time during the network growth, the system displays fundamentally different behavior that may be termed emergent behavior
4	Emergence of hubs and clusters	Networks starts displaying small-world effects and a network hierarchy composed of hubs and clusters emerges
5	Power-law behavior	Some agent/systems become network enablers and facilitate "long" weak-ties that sustain large-scale topologies
6	Nonlinear interactions	System self-organized through these hubs result in emergent interactions that brings new affordances and/or constraints
7	Preferential attachment	Agents/systems exhibit affinity for other agent/systems that changes the inherent structure itself
8	Vulnerability to attacks if targeted to hubs	Any attack on the hubs results in cascaded effects and push the entire system toward self-organized criticality when it reaches a critical state
9	Threshold levels	Each agent/system has a threshold model that controls its preferential attachment
10	Concurrency and multi-tasking	Each agent/system is modular and has defined interfaces and capabilities that are operational in a concurrent manner

Barabasi (2003) and Newman (2001). Mittal (2013a) summarized the structural scale-free characteristics of CAS, as enumerated in Table 6.1.

If the problem was to just address the scale-free nature with homogeneous agents/systems, there are ample tools that can model a scale-free structure. However, any real-world CAS is a heterogeneous system with multiple characteristics. Modeling such a heterogeneous system requires modeling agent characteristics dependent on the source of behavior for that agent, which may be domain dependent. The complexity increases when the model has multi-level specifications, i.e., a containment hierarchy exists. The problem further grows exponentially if the agent/system, even though available as a modular component, is modeled in a dissimilar paradigm, such as discrete, continuous or hybrid. Undoubtedly, in such a situation, the presence of emergent behavior is but natural.

The primary question is:

> Do we know enough about the system to recognize the emergent behavior and ensure that it is the right one, to begin with?

Mittal and Zeigler (2017) stated that multi-paradigm modeling is the preferred means to bring in domain knowledge from multiple disciplines to develop a CAS model. Some of the disciplines they enumerated are: Cognitive Psychology, Network Science, Human Factors, Communication Systems, Ontology, Complexity Science, Supply-Chain, and Power Systems. Each of the disciplines contributes toward defining the behavior of an agent/system and provides constraints to control the structure of the resulting CAS. Further, an agent may have a variable degree of autonomy and in contrast, a system may be a passive system with limited autonomy.

6.3 Adaptive Aspect in CAS Modeling

From the behavioral perspective, CAS are systems that display two primary behaviors: adaptive behavior and emergent behavior. Let us look at the modeling of these two behavioral aspects.

Adaptive behavior manifests at the agent–environment boundary when the agent situates itself in an environment. This agent–environment interaction results in changing the agent's original behavior as it continues to adapt to the environmental conditions. An *adaptive system* is a system that changes in the face of perturbations to maintain invariant state by altering its properties or modifying its environment. The ability to adapt depends on the observer who chooses the scale and granularity of description. An adaptive system is necessarily complex, but the opposite is not necessarily true.

Evolution is a result of an adaptive system. In fact, John Henry Holland (1992) was one of the biggest contributors that led to the inception of the field from the perspective of adaptation. He introduced the concept of genetic evolution in describing adaptive systems. He was interested in the question of how computers could be programmed so that problem solving capabilities are built by specifying *what to be done*, instead of *how to do it* (Brownlee 2007). Holland conceptualized an adaptive plan, which was the continuous modification of structures by means of genetic operators. The specialization of this adaptive plan, called the genetic plan, represented at the end, laid the foundation of the field of genetic algorithms and, more generally, evolutionary computation.

The design of an adaptive system that manifests emergent behavior becomes a complex and challenging task (Branke and Schmeck 2008). It is impossible to know how design choices made at a component level affect the overall system behavior. Thus, these choices must be evaluated by means of computationally extensive models and the corresponding simulations. As stated before, adaptability is based on evolution. Thus, the adaptive behavior of these models can be tackled using evolutionary computation. Figure 6.1 depicts an illustrative example of how the CAS behavior can be modeled through evolutionary computation.

Evolutionary Algorithms (EAs) make them excellent candidates for dealing with M&S of CASs (Alba and Cotta 2002; Branke 2001; Deb 2001; Arnaldo et al. 2013).

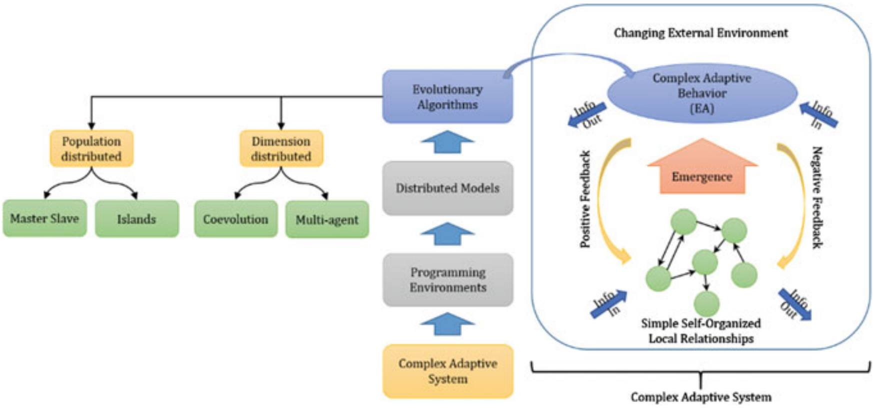


Fig. 6.1 Evolutionary algorithms in CAS modeling

First, EAs act as a black box in an optimization heuristic (the partial search algorithm). There are no restrictions on the fitness function. Thus, these algorithms can be used in complex simulation environments. Second, the fitness evaluation can often be simplified by means of the reduction of the computational complexity. This fact allows us a simplification of the initial complex environment. Third, EAs behave well with uncertainty in evaluation, can manage adaptation to changing or adaptive environments, as well as in real-time scenarios. In this regard, EAs are considered adaptive and *interactive*. This last feature makes them useful in the study or design of desired emergent behavior. Fourth, there exist a group of EAs called multiobjective evolutionary algorithms (MOEAs).

These methods are designed to deal with multiple objectives or fitness functions. This is a realistic method to quantify emergent behavior that looks for different alternatives, depending on the chosen objective in each situation. Using relations like Pareto dominance or Nash ascendancy, these algorithms can offer a good spectrum of different solutions to the problem under consideration (Deb et al. 2005). Finally, from a performance perspective, EAs are easily parallelizable, even on heterogeneous hardware platforms or in cloud environments. Being based on the evolution of a population, the repetitive evaluation of each individual in the population can be distributed using a master–worker or an islands-based distribution. Coevolution and multi-agent systems can also be used in the distribution of EAs, as Fig. 6.1 shows.

There are many examples in the literature of modeling complex adaptive systems and inducing desired emergent behavior. To name but a few, Chellapilla and Fogel (2001) used a genetic algorithm to evolve neural networks able to play the game of checkers. The major achievement of this work is based on the competitive strategy, which is evolved given only the spatial positions of pieces on the board and the piece differential, a decision approach that would normally require human input and expertise. Andre and Teller (1999) successfully applied evolutionary methods to develop a program to control a team of robot soccer players. The evolutionary algorithm operated with a set of basic control functions such as turning, running, and kicking. The fitness function was global, evaluating a general good play. There was no score of specific tasks like tracing the ball, kicking in the correct direction, goal scoring, etc. The robot team, named Darwin United, entered in the international RoboCup annual tournament and competed with other teams of autonomous robots (RoboCup 2017). Darwin United outranked half of the human written, highly specialized programmed teams, performing quite well.

6.4 Emergent Behavior Aspect in CAS Modeling

Emergence has been a native of the land of "complex systems" and there are four schools of thought that study emergence, as summarized by Wolf and Holvoet (2005):

- *Complex adaptive systems* theory: Concept of macro-level patterns arising from interacting agents.
- *Nonlinear Dynamical Systems theory* and *Chaos theory*: Concept of attractors that guide the system behavior.
- *The synergistic school*: Concept of order parameter that influences which macro-level phenomena a system exhibits.
- *Far-from-equilibrium thermodynamics*: Concept of dissipative structures and dynamical systems arising from far-from-equilibrium conditions.

New fields of application of emergence are System of Systems (SoS) (Maier 1998; Mittal and Rainey 2015) and complex sociotechnical systems (Mittal 2014). Modeling and simulation support to SoS engineering (Rainey and Tolk 2015) discusses four types of emergence, extended from the works of Maier (1998, 2015):

- *Simple emergence*: The emergent property or behavior is predictable by simplified models of system's components.
- *Weak emergence*: The emergent property is readily and consistently reproduced in simulation of the system but not in reduced complexity nonsimulation models of the system, i.e., simulation is necessary to reproduce it.
- *Strong emergence*: The emergent property is consistent with the known properties, but even in simulation is inconsistently reproduced without any justification of its manifestation.
- *Spooky emergence*: The emergent property is inconsistent with the known properties of the system and is impossible to reproduce in a simulation of a model of equal complexity as the real system.

Weak emergent properties can be formally specified using mathematical principles and reproduced in a simulation environment (Mittal 2013a; Szabo and Teo 2015). They are known a priori. In contrast, Strong emergent properties are

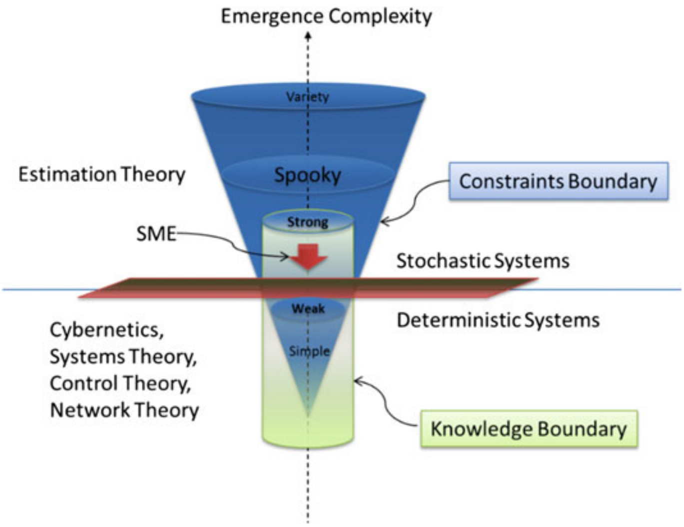


Fig. 6.2 Emergence complexity cone (Mittal and Rainey 2015)

discovered, after the fact. Strong emergent behavior is bounded by the knowledge of the existing properties of the system. Consequently, new knowledge is always generated when strong emergent properties are known for the first time (Mittal 2013a). When they are known during a simulation study by stochastic methods (Mittal and Rainey 2015) and are validated as a new property that is consistent with the known properties by the system's subject matter experts (SMEs), they are then incorporated as "new behavior" that is not "emergent" anymore. Once known they can be now formalized as weak emergent behavior. This results in a cyclical process of knowledge discovery and knowledge augmentation in the existing model (Mittal 2014), and can now be consistently reproduced in a simulation.

Stochastic studies and uncertainty quantification lie in the simulation domain. These computational methods used in conjunction with the existing theories help validate the existing theories or suggest modifications to them. These also help define the boundaries of the theory, in turn, giving us specific knowledge about the constraints and limitations of that theory which must be implemented for understanding the emergent behavior. Figure 6.2 shows the Emergence Complexity Cone linking emergent behavior taxonomy in increasing complexity on the y-axis, and division of the cone into stochastic and deterministic search-spaces on the x-axis. The deterministic domain is supported by established theories, like cybernetics, Systems theory, control theory, and network theory. The stochastic domain is supported by estimation theory. Cone volume depicts the variety[1] (Ashby 1956). Cone perimeter depicts constraints, and the knowledge boundary as a cylinder that

[1]Variety refers to the total number of states in the system.

addresses the variety and constraints. The knowledge cylinder around simple and weak emergence in the deterministic domain signifies ample knowledge available to develop abstractions. A diverging cone reflects the increasing complexity as constraints are loosened in the stochastic domain leading to an increase in variety and lack of theoretical constructs to understand the overall complex behavior.

Further, computationally, the emergent behavior can be understood using today's latest technology of feature engineering, deep learning, and data analytics using big data and machine learning (Tolk 2015). This would facilitate the identification of various holistic behaviors and their classification that would aid designing various algorithms for detecting the eventual emergent behavior that may be useful in SoS context. This aspect is largely bottom-up. Once various observers are computationally made available, they can be integrated in the agent behavior repertoire so that the emergent behavior, that is now detectable and perceivable at the agent–environment boundary through various newly developed sensors (both software and hardware), can be used and acted-upon by the agent. This process refines the agent behavior with respect to the environment it is in, thereby situating it in the environment, and hence manifesting adaptable behavior. This activity now becomes top-down as there is a conscious design process (done by a human) that is employed for such behavior refinement and incorporating of new emergent behavior as a causal behavior impacting the existing behavior of the agent. The process thus becomes increasingly cyclical, wherein simulation studies are constantly discovering new consistent emergent behaviors that are then continuously added to the agent/system behavior repertoire.

We shall now see how data is an integral part of the simulation-based CAS engineering process.

6.5 Data-Driven Analytics for CAS Model Evolution

Complex systems are dynamic in nature and usually present failures in real time that their controller must be able to manage. Given an IoT system, for example, we may observe how sensors are continuously failing, but the complex system must be able to adapt itself to the new situation. In these new kinds of problems, a global and adaptive modeling and optimization strategy can be especially useful. These systems are highly complex and cannot be easily tackled using classic modeling, simulation, and optimization techniques. They demand a heterogeneous and multi-level approach. For this purpose, the M&S work starts with a precise knowledge of the problem, developing techniques, and methodologies that begin with data acquisition in real environments, and observing the practical constraints of the given scenario.

Figure 6.3 shows an overview of the data-driven methodology. With an initial raw dataset, a MBSE process starts. To this end, the minimal set of features used to build the model is initially selected. This model is then simulated for verification

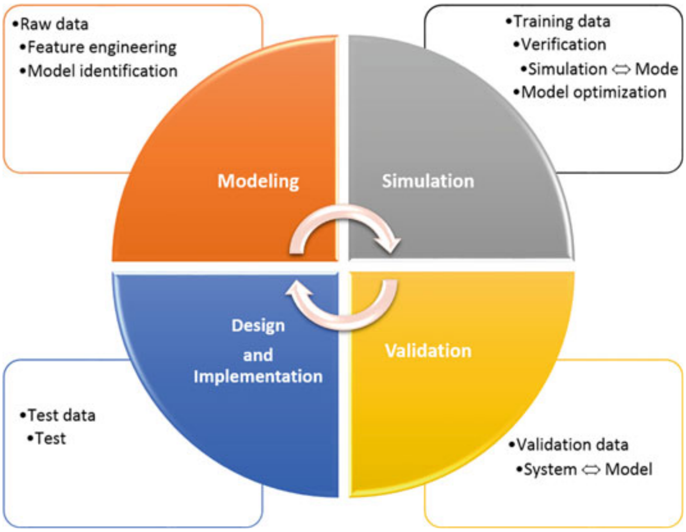


Fig. 6.3 Data-driven M&S in CAS

and optimization. Finally, the model is validated and tested and the actual system is implemented (Zeigler et al. 2000).

Using the above methodology for CAS engineering is a challenge. One of the major problems in the methodology detailed above is the constant necessity of human experts to take certain decisions in the modeling part, mainly because of the unpredictability of the emergent behavior in CAS. To tackle this issue, and as stated in Sect. 6.3, the model is usually complemented with an evolutionary computation module to emulate this emergent behavior. Some algorithms frequently used are particle swarm optimization (PSO) (Zhang et al. 2017) or ant colony optimization (ACO) (Wang et al. 2016) that emulate emergent features to solve complex optimization problems. Evolutionary computation can reproduce complex adaptive models.

However, this is not enough. Although the aforementioned models faithfully reproduce complex, dynamic and "alive" systems, the limitation is always found in the heuristic used by the corresponding meta-heuristic. So, the question is:

> Is there always the need for human experts in the process of modeling complex optimization heuristics?

The answer is no; we may find several approaches to turn these heuristics into hyper-heuristics (Zheng et al. 2015). These methods try to adapt various low-level heuristics into a higher level hyper-heuristic. This task is usually performed using machine-learning techniques. This technique is very useful because it is implemented under the assumption that once the model is integrated into the final system; it may also evolve. This evolution highly depends on the input data, as well as on the number of features. The set of features can also change during the whole model

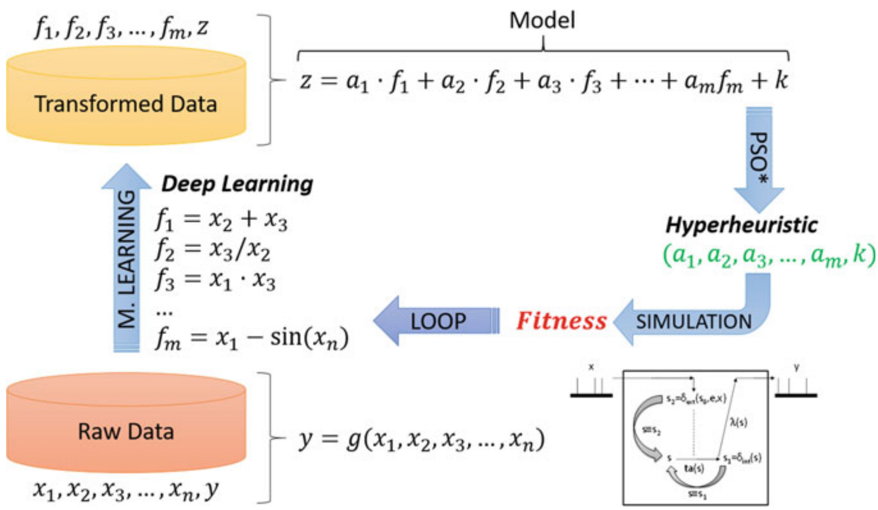


Fig. 6.4 Specific view of the CAS modeling part

life cycle. Regardless of the machine-learning approach used (e.g., Feature engineering techniques, Deep Learning, pattern-matching, etc.), these techniques usually work in one of the following directions: generating new heuristics from basic sub-structures, automatically managing the search operators in the main evolutionary algorithm, or tuning the control parameters of the algorithm itself.

Figure 6.4 shows a specific view of the aforementioned CAS modeling part and the computational support as a part of model evolution process.

As the first step, it is required to have a representative data set of the system for which the model will be defined. Next, using machine-learning algorithms (like Deep Learning) and in combination with the hyper-heuristic approach, the set of features or input variables that are representative for the model are obtained, as well as the model itself, which is a function of these features. Feature engineering is a particularly useful technique to select an optimal set of features that best describe a complex model. Those features consist of measurable properties or explanatory variables of a phenomenon (Arroba et al. 2014).

Second, the first version of the model is simulated. The system's behavior is studied, simultaneously verifying and validating the reliability of the system both in virtual and in real-time environments. Often, the provided model or even the modeled system requires an optimization cycle. Minimization of risks, cost, as well as maximization of performance are some examples. A varied set of optimization techniques can be applied for this purpose: MILP (mixed integer linear programming), simulated annealing, genetic algorithms, particle swarm optimization, genetic programming, multiobjective optimization, etc. The model obtained is more robust and contextualized with real-data incorporated through the machine learning and hyper-heuristic phases. Finally, as Fig. 6.3 depicts, the system is implemented. This implementation is performed systematically. In hardware–software co-design methodology that has M&S as an integral part of systems engineering life cycle, hardware components are first virtualized as software models and then iteratively replaced by hardware-in-the-loop simulation. In a hardware–software co-simulation environment (Mittal et al. 2015; Risco-Martin et al. 2016), the target system is verified and validated using all the previous procedures and the general structure shown in Fig. 6.3.

6.6 Co-simulation Environment for CAS

Designing and evaluating a CAS is a three-step process (Clymer 2009). First, a mathematical model is required to define and represent a CAS environment with precision. It describes the set of agent interactions that happen between the CAS agents during the system's operation. Second, a modeling language that implements a set of equations for modeling dynamical behavior of the agent and the environment. The modeling language, as a design workbench, should have both textual and graphical elements that enhance the visualization and specification of CAS structure and behavior. Finally, an evaluation tool that ensures that the application of Systems Theory, expressed using the aforementioned graphical modeling language still holds. As we have discussed earlier, a CAS model is a multi-domain model. Consequently, constructs from multi-paradigm modeling should be adhered to.

A multi-paradigm modeling (Vangeluwe and Lara 2003) aligns different paradigms with their corresponding modeling formalisms and implementation types toward a composite model capable of exchanging information across various abstraction levels.

Table 6.2 (Mittal and Zeigler 2017) enumerates some of the tools that are currently used in a particular domain. Some tools are language dependent (e.g., C ++, Java, LISP, and DSL) and/or some are platform dependent (e.g., Windows, Linux,

Table 6.2 Domains and their tools (not a complete list) (Mittal and Zeigler 2017)

ID	Domain	Modeling tool/Architecture
1	Cognitive psychology	ACT-R (2017), jACT-R (2017), SOAR (2017
2	Network science	Gephi (2017), NetworkX (2017), IGraph (2017), Statnet (2017), Pajek (2017)
3	Human factors	JACK (2017), Kinemation (2017)
4	Communication systems	OpNet (2017), OmNet (2017), NS3+ (2017), MATLAB/Simulink (2017)
5	Ontology	Protege (2017), TwoUse (2017), NeOn (2017), FlexViz (2017)
6	Complexity science	NetLogo (2017), RePast/Symphony (2017), DEVS (Zeigler et al. 2000), R (2017)
7	Supply-chain	MS Excel, Arena, SAS
8	Power systems	GridLab-D (2017)

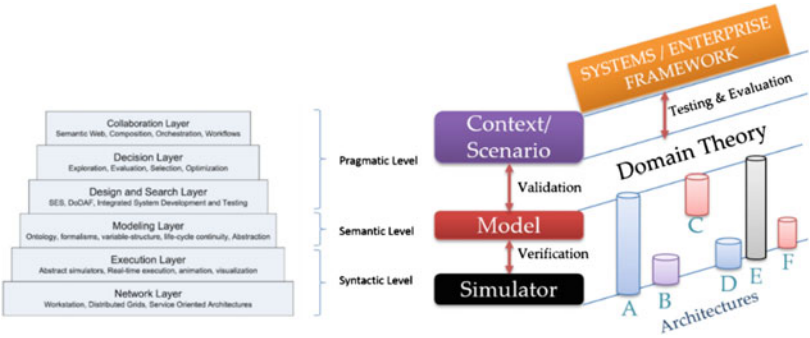


Fig. 6.5 M&S with verification and validation and T&E (Mittal 2014)

and Mac). Some are discrete event; some are continuous and closed-form. These tools have their own software architecture, subscribe to a scientific theory and sometimes the software is proprietary. Figure 6.5 shows A, B, C, D, E, and F as sample architectures. It also shows a layered M&S architecture addressing the pragmatic, semantic, and syntactic levels of interoperability. For more details, see Mittal et al. (2008).

Developing a modeling workbench for a multi-paradigm modeling environment is a nontrivial exercise and two solutions exist. Given a set of modeling formalisms that a CAS model needs, the first option requires the use of a formalization transformation graph (Vangheluwe 2000) that transforms a relatively simpler formalism to be transformed into a more rigorous formalism. Such was an approach used by AtoM3 (Vangheluwe and Lara 2003). However, one then must understand each of the other formalisms and their semantics to ensure that there are no leaky abstractions and the mapping is correct. The other approach is to transform these modeling formalisms to hybrid super-modeling formalism that can model both discrete and continuous systems at the fundamental level, for example, DEV&DESS (Cellier 1977; Praehofer 1991; Zeigler et al. 2000; Mittal and Martin 2013). This would ensure that the appropriate abstractions from each of the formalisms are integrated at the state-event and time-event levels and are simulated in a mathematically verifiable way, as implemented in the DEVS formalism (Zeigler et al. 2000).

Bringing various simulators together is more than a typical software engineering integration exercise. The relationship between the modeling formalism and the underlying simulator is sacrosanct and unless it is rigorously implemented, will probably yield emergent behaviors. The source of such emergent behavior then is not the lack of knowledge to model, but failure to implement the simulator, which now has become a multi-simulator. Without the DEVS super-formalism as a foundation (Mittal and Zeigler 2017) that specifies an abstract simulation protocol between the model and the simulator (Zeigler et al. 2000), it would resemble engineering a multi-threaded software program with no verifiable inter-thread communication protocols to guarantee timeliness and accurate concurrent execution in an SoS setting. The task of integrating various simulators to perform together as a composite simulation is termed as co-simulation. This involves weaving the time series behavior and data exchanges accurately, failure of which, will yield inaccurate simulation results. Every such hybrid system would require a dedicated effort to build a co-simulation environment. Earlier work on DEVS-Bus (Kim and Kim 1998), the netcentric SOA simulation infrastructure (Mittal and Martin 2013), and recent work in building cyber-physical simulation environments (Lee et al. 2015), agent-based power-hardware-in-the-loop with simulation infrastructure at National Renewable Energy Lab, Dept. of Energy, USA (Mittal et al. 2015; Pratt et al. 2017) along with a multi-agent toolkit MECYSCO (Camus et al. 2017) provide a solid foundation to execute a hybrid discrete event complex continuous model in a parallel distributed co-simulation environment using a super-formalism such as DEVS formalism, producing accurate results. Earlier efforts at Oak Ridge National Lab (Nutaro et al. 2008) and usage of agent-based co-simulation (Kilkki et al. 2014), both for Smart Grid M&S, provide further evidence. The integrated model can be used for combined simulation of electrical, communication, and control dynamics.

When the simulator code is open-source or available, the architecture of the tools could be understood and interventions can be made to weave the external tool's input. However, many times these simulation tools are proprietary architectures with no access to the code-base. In these cases, there is truly no means to verify the model's execution at the simulator level. At this point, the emergent behavior, which now is a function of computational engineering in the simulation environment, cannot be overcome. In other words, even though model representation is valid, it cannot be verified computationally. Without a robust multi-simulation or co-simulation environment, advances in CAS will hit a ceiling and any progress on modeling will not achieve desired and reproducible results.

6.7 Verification and Validation of CAS Models

Validation is the process of ensuring that a model is a reasonable representation of a real-world system. To validate, input and output trajectories between the source system (whether real or conceptual) and the model under validation must be generated. Validity, whether replicative, predictive, or structural requires that these trajectories be equal (Zeigler et al. 2000). *Verification* is the attempt to establish that the *simulation relation* holds between a simulator and a model, i.e., the simulator faithfully implements the model's dynamic behavior. There are two general approaches to verification: formal proof of correctness and extensive testing (Zeigler et al. 2000). This is in congruence with the ideas of Robert Sargent (2011) that links data validity as a central concept linking the conceptual model, the computerized model and the problem entity (a.k.a real-world "system"). Per Sargent, the relationship between the real-world system and its conceptual model is called conceptual model validation, while the relationship between the system and the computerized model is called operational validation (that is support by a computational execution of the model). This is the *modeling relation* per Zeigler's definition. The relationship between the conceptual model and the computerized model is identified as computerized model verification. Rightly so, per Zeigler, this is the simulation relation that ensures a model is implemented correctly in a simulation environment and the entity (simulator) that ensures that a strict relation (i.e., simulation relation) exists between the conceptual model and the computerized model. Bair and Tolk (2013) summarized various definitions toward a unified theory of validation. While model validation is user-faced, verification is implementation-specific (Mittal and Zeigler 2017). The path to verification of simulation models is not straightforward and a huge gap exists. Formal methods for model verification is an active area of research (Gore and Diallo 2013; Tolk et al. 2013).

In Sect. 6.6, we mentioned how the proposed CAS-based M&S methodologies are emerging in current real-world applications. However, these M&S techniques remain difficult to verify and validate (V&V). Performing V&V is an exhaustive exercise for any simulation model. Due to inherent complexity in CAS simulation that comprise of a heterogeneous agent-based model wherein the behavior of each agent is adaptive, V&V is a challenge of its own. The potential complexity to this issue does not stop here. CAS models usually contain many variables, so there exists a prominent risk of over-fitting in the process of feature selection. Additionally, because CAS simulation is stochastic, a single run is insufficient to verify or to validate the quality of model parameters. Numerous runs are required to build confidence in the simulation results. Last but not the least, are the consequences of emergent behaviors: emergent properties of CAS V&V cannot be easily expressed. Thus, current standardized and formalized V&V methodologies will need to be modified and adapted to incorporate an evolutionary V&V framework for integrated CAS testing and evaluation (T&E).

For CAS models, the validation aspect answers the question about how useful a model is in a scenario. The verification aspect falls-back to the co-simulation of various tools and paradigms that need to be brought in for an accurate simulation of a model. The entire simulation experiment, the model and the simulation infrastructure must be automated through a model-based repository and transparent simulation framework.

Per Mittal (2014), validation is the only aspect that can be ensured in CAS M&S. Verification must be ascertained using statistical and stochastic methods. Rouff et al. (2012) proposed the following verification methodology:

- A stability analysis capability that identifies instabilities given in a system model and partitions the system model into stable and unstable component models.
- A state space reduction capability that prunes the state space of an unstable component model without loss of critical fidelity.
- High-performance computing simulations to explore component behavior over a wide range of an unstable component's reduced state space and produce a statistical verification for the component.

- A compositional verification capability that aggregates individual component verification.
- Operational monitors to detect and act to correct undesired unstable behavior of the system during operation.

However, we may also formulate the hypothesis that, inside a complex stochastic system, the V&V techniques should be stochastic as well. This would allow us the exploration of new combinations of model parameters as good candidates to reduce the divergence between the simulation and the real-world data. This process can be formulated through a deep learning approach at the metamodel layer over the CAS simulation. Indeed, this is an opportunity that we must avail in our next article.

6.8 Conclusions and Future Work

Complex adaptive systems manifest emergent behavior in a dynamic environment. While there are many options available for CAS modeling, without a simulation-based approach, models cannot be verified and experimented in an exhaustive and stochastic manner. Paradigms like MBSE, MDE, MBE which support traditional systems engineering need to be augmented with simulation-based methodologies to ensure they support complex systems engineering that integrate discrete and continuous systems for complex hybrid systems. This then needs to be augmented with evolutionary computation techniques to incorporate adaptive and emergent behaviors in a computational environment for large-scale experimentation, testing, and evaluation.

CAS model complexity at the structural level must be studied using Network Science to ascertain the impact a connected environment has on agents/system situated in that environment. This is essential as the structure of the environment yields the overall behavior of CAS. Any change in the structure of the overall system resulting from behavior of an individual agent/system is a critical event that results in changing the behavior of agents/systems in that agent's neighborhood or far across the network through weak-ties. Indeed, a new world that is replete with cyber-physical systems and Internet of Things is a complex world.

The adaptive nature of CAS needs to be modeled using evolutionary computation techniques such as genetic algorithms, where in, global fitness functions reuse basic behavior building blocks to yield a behavior that is more in tune with the environment, thereby, situating the agent/system in the environment. This can only be achieved in a computational environment using simulation-based methodologies.

Emergent behavior is an essential element in any CAS study. While desired emergent behavior can be modeled in a top-down manner, computational environment becomes a necessity when bottom-up behavior needs to be studied and evaluated. Various agent-based modeling tools are available that can be used to develop an abstract model of CAS. The formalization of emergent behavior must be supported with fundamental scientific theories such as cybernetics, systems theory, network science, control theory, and estimation theory such that the observed emergent behavior in a computational environment is consistent with the known theories as well as with the system's domain of application. Any inconsistent emergent behavior should then be rejected based on such evidence providing a learning opportunity to perform model correction or simulator correction. Both model correction and simulator correction are nontrivial endeavors when the model is a multi-paradigm model and the simulator is a co-simulator that incorporates multi-simulation. A multi-paradigm model should conform to a super-formalism such as DEVS formalism that can model various discrete event and complex continuous hybrid formalisms and adhere to systems theory thereby, yielding guaranteed emergent behavior. Having ensured that, the focus next is on the model-simulator relationship. The co-simulation environment must preserve this relationship to eliminate emergent behaviors arising out of simulator engineering.

Data-driven methodology is another essential element in any CAS study. The support given by a correct M&S must be augmented with a data strategy as it helps validation and verification of the model as well as of the simulator itself. The empirical data guides the model formulation. Various heuristics, meta heuristics, and hyper-heuristics are then designed as algorithms, which are then computationally implemented in an agent-based simulation environment. The simulation experiments yield simulation data that is then compared with raw data for evaluating model's validity. Any deviation between these two datasets is then treated as an opportunity to improve the model, as well as of heuristics. By the definition of emergent behavior, these macro-behaviors are sometimes irreducible at the micro-level. Consequently, the integration of data-driven methodology is essential to fine-tune both the model and heuristics as it is unknown where to make the correction at the agent level. These heuristics represent macro-level algorithms that act on a group of agents/systems. Once this cyclical process is implemented and is supported by a robust co-simulation environment, meaningful CAS engineering can be attempted.

CAS engineering is a complex endeavor requiring professionals from multiple disciplines. The subject matter expertise of the agent, the systems, and the operational environment should be translated into valid models in a verifiable simulation environment. This requires a partnership between domain experts, modeling experts, and simulation infrastructure experts. CAS engineering will not become possible unless the undesired emergent behaviors are completely removed from a computational environment or are known a priori so that they can be knowledgeably eliminated. A computational simulation-based environment provides experimentation opportunities to validate a CAS model, such that it becomes predictable and eventually useful. Only when a model becomes valid and predictable, real-world systems engineering can begin.

Review Questions

1. What is the difference between complicated and complex systems?
2. What theoretical background do you need to understand both complex and adaptive systems?
3. What is the difference between a complex adaptive system and system of systems?
4. When does a system of systems becomes a complex adaptive system?
5. How can you incorporate emergent behavior in engineering complex adaptive systems?
6. Why is a simulation environment engineering a critical component in engineering complex adaptive systems?
7. Which is the role of evolutionary computation in the design of complex adaptive systems?
8. Is there always the need for human experts in the process of modeling complex optimization heuristics?
9. What is the difference between model verification and model validation? Can we ensure verification and validation in CAS models?

References

Adaptive character of though-rational (act-r). (2017). http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/. ([Online; accessed 20-Feb-2017]).

Alba, E., & Cotta, C. (2002). Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 6, 443–462.

Alexander, C. (1964). Notes on the synthesis of form. Harvard University Press.

Andre, D., & Teller, A. (1999). Evolving team Darwin United. In M. Asada & H. Kitano (eds.), Robocup-98: Robot soccer world cup ii (pp. 346–351). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/3-540-48422-1 28.

Arifin, S. M. N., & Madey, G. R. (2015). Verification, validation, and replication methods for agent-based modeling and simulation: Lessons learned the hard way! In L. Yilmaz (Ed.), Concepts and methodologies for modeling and simulation: A tribute to Tuncer Ören (pp. 217–242. Springer International Publishing. doi:10.1007/978-3-319-15096-3 10.

Arnaldo, I., Cuesta-Infante, A., Colmenar, J. M., Risco-Martin, J. L., & Ayala, J. L. (2013). Boosting the 3D thermal-aware floorplanning problem through a master-worker parallel MOEA. Concurrency and Computation: Practice and Experience, 25 (8), 1089–1103. doi:10.1002/cpe.2902.

Arroba, P., Risco-Martin, J. L., Zapater, M., Moya, J. M., & Ayala, J. L. (2014). Enhancing Regression Models for Complex Systems Using Evolutionary Techniques for Feature Engineering. Journal of Grid Computing, 13 (3), 409–423. doi:10.1007/s10723-014-9313-8.

Ashby, W. (1956). An introduction of cybernetics. Chapman and Hall.

Bair, L. J., & Tolk, A. (2013). Towards a unified theory of validation. In Proceedings of the 2013 winter simulation conference: Simulation: Making decisions in a complex world (pp. 1245–1256). Piscataway, NJ, USA: IEEE Press.

Barabasi, A. (2003). Linked: How everything is connected to everything else and what it means for business, science and everyday life. Penguin Books.

Branke, J. (2001). Evolutionary optimization in dynamic environments. Kluwer Academic Publishers.
Branke, J., & Schmeck, H. (2008). Organic computing. In (p. 123–140). Springer Berlin Heidelberg.
Brownlee, J. (2007). Complex adaptive systems (Tech. Rep.). Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information Communication Technology, Swinburne University of Technology, Melbourne, Australia.
Camus, B., Paris, T., Vaubourg, J., Presse, Y., & Bourjot, C. (2017). MECYSCO: a multi-agent DEVS wrapping platform for the co-simulation of complex systems [research report] (Tech. Rep.). ORIA, UMR 7503: CNRS.
Cellier, F. E. (1977, September). Combined continuous/discrete system simulation languages: Usefulness, experiences and future development. SIGSIM Simul. Dig., 9 (1), 18–21. doi:10.1145/1102505.1102514.
Chellapilla, K., & Fogel, D. B. (2001, Aug). Evolving an expert checkers playing program without using human expertise. IEEE Transactions on Evolutionary Computation, 5 (4), 422–428. doi:10.1109/4235.942536.
Clymer, J. R. (2009). Simulation-based engineering of complex systems. Wiley.
Deb, K. (2001). Multi-objective optimization using evolutionary algorithms.Wiley.
Deb, K., Mohan, M., & Mishra, S. (2005). Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of paretooptimal solutions. Evolutionary Computation, 13 (4), 501–525. doi:10.1162/106365605774666895.
Flexviz. (2017). https://sourceforge.net/projects/flexviz/. ([Online; accessed 20-Feb-2017]).
Gephi. (2017). http://gephi.org. ([Online; accessed 20-Feb-2017]).
Gore, R., & Diallo, S. (2013). The need for usable formal methods in verification and validation. In Proceedings of the 2013 winter simulation conference: Simulation: Making decisions in a complex world (pp. 1257–1268). Piscataway, NJ, USA: IEEE Press.
Gridlab-d. (2017). http://www.gridlabd.org. ([Online; accessed 20-Feb-2017]).
Holland, J. (1992). Complex adaptive systems. Daedalus, 121, 17–30.
Igraph. (2017). http://igraph.org/redirect.html. ([Online; accessed 20-Feb-2017]).
Jack. (2017). https://www.plm.automation.siemens.com/en us/products/tecnomatix/manufacturing-simulation/human-ergonomics/jack.shtml. ([Online; accessed 20-Feb-2017]).
Java act-r. (2017). http://jact-r.org/. ([Online; accessed 20-Feb-2017]).
Kilkki, O., Kangasrääsiö, A., Nikkilä, R., Alahäivälä, A., Seilonen, I. (2014). Agent-based modeling and simulation of a smart grid: A case study of communication effects on frequency control. Engineering Applications of Artificial Intelligence, 33, 91–98. doi:10.1016/j.engappai.2014.04.007.
Kim, Y., & Kim, T. (1998). A heterogeneous simulation framework based on the DEVS bus and the High Level Architecture. In (Vol. 1).
Kinemation. (2017). http://www.cs.cmu.edu/~german/research/HumanApp/humanapp.html. ([Online; accessed 20-Feb-2017]).
Lee, E. (2008). Cyber physical systems: Design challenges. In IEEE international symposium on object oriented real-time distributed computing.
Lee, K., Hong, J., & Kim, T. (2015). System theoretic formalisms for combined discrete-continuous system simulation. ETRI Journal, 37.
Maier, M. (1998). Architecting principles for system-of-systems. Systems Engineering, 1, 267–284.
Maier, M. (2015). The Role of Modeling and Simulation in system of systems development, in L. B. Rainey, A. Tolk (eds.), Modeling and Simulation Support for System of Systems Engineering Applications, Hoboken, NJ: John Wiley and Sons.
Matlab/simulink. (2017). https://www.mathworks.com/products/simulink.html. ([Online; accessed 20-Feb-2017]).
Miller, J. H., & Page, S. E. (2009). Complex adaptive systems: An introduction of computational models of social life. Princeton Press.

Mittal, S. (2013a). Emergence in stigmergic and complex adaptive systems: A formal discrete event systems perspective. Cognitive Systems Research, 21, 22–39.
Mittal, S. (2013b). Netcentric complex adaptive systems. In Netcentric systems of systems engineering with DEVS unified process. CRC Press.
Mittal, S. (2014). Model engineering for cyber complex adaptive systems. In European modeling and simulation symposium.
Mittal, S., & Martin, J. L. R. (2013). Netcentric system of systems engineering with DEVS unified process. Boca Raton, FL, USA: CRC Press.
Mittal, S., & Rainey, L. (2015). Harnessing emergence: The control and design of emergent behavior in system of systems engineering. In Proceedings of summer computer simulation conference. SCS.
Mittal, S., Ruth, M., Pratt, A., Lunacek, M., Krishnamurthy, D., & Jones, W. (2015). A system-of-systems approach for integrated energy systems modeling and simulation. In Summer computer simulation conference.
Mittal, S., & Zeigler, B. P. (2017). Theory and practice of M&S in cyber environments. In A. Tolk & T. Ören (Eds.), The Profession of Modeling and simulation. Wiley & Sons.
Mittal, S., Zeigler, B. P., Martin, J. L. R., Sahin, F., & Jamshidi, M. (2008). Modeling and simulation for systems of systems engineering. In System of systems engineering (pp. 101–149). John Wiley & Sons, Inc. doi:10.1002/9780470403501.ch5.
Neon toolkit. (2017). http://semanticweb.org/wiki/NeOnToolkit.html. ([Online; accessed 20-Feb-2017]).
Netlogo. (2017). https://ccl.northwestern.edu/netlogo/. ([Online; accessed 20-Feb-2017]).
Networkx. (2017). https://networkx.github.io/index.html. ([Online; accessed 20-Feb-2017]).
Newman, M. (2001). Clustering and preferential attachment in growing networks. Physical Review, 64.
Ns3. (2017). https://www.nsnam.org. ([Online; accessed 20-Feb-2017]).
Nutaro, J., Kuruganti, P. T., Shankar, M., Miller, L., & Mullen, S. (2008). Integrated modeling of the electric grid, communications, and control. International Journal of Energy Sector Management, 2, 420–438.
Omnet. (2017). https://omnetpp.org. ([Online; accessed 20-Feb-2017]).
Opnet modeler. (2017). https://www.riverbed.com/products/steelcentral/opnet.html. ([Online; accessed 20-Feb-2017]).
Pajek. (2017). http://mrvar.fdv.uni-lj.si/pajek/. ([Online; accessed 20-Feb-2017]).
Plsek, P. E., & Greenhalgh, T. (2001). The challenge of complexity in health care. BMJ, 323 (7313), 625–628. doi:10.1136/bmj.323.7313.625.
Praehofer, H. (1991). System theoretic formalisms for combined discrete continuous system simulation. International Journal of General Systems,19, 226–240.
Pratt, A., Ruth, M., Krishnamurthy, D., Sparn, B., Lunacek, M., Jones, W., Mittal, S., Wu, H., Marks, J. (2017). Hardware-in-the-loop simulation of a distribution system with air conditioners under model predictive control. In IEEE Power Engineering Society General Meeting.
Protege. (2017). http://protege.stanford.edu. ([Online; accessed 20-Feb-2017]).
Rainey, L., & Tolk, A. (Eds.). (2015). Modeling and simulation support for system of systems engineering applications. Hoboken, NJ, USA: John Wiley and Sons.
Repast/symphony. (2017). https://repast.github.io/repastsimphony.html. ([Online; accessed 20-Feb-2017]).
Risco-Martin, J. L., Mittal, S., Fabero, J. C., Malagfion, P., & Ayala, J. L. (2016). Real-time Hardware/Software Co-Design Using DEVS-based Transparent M&S Framework. In Proceedings of the 2016 summer simulation multiconference (Summersim'16), Chicago, IL.
Robocup. (2017). http://www.robocup.org. ([Online; accessed 23-Feb-2017]).
Roufi, C., Buskens, R., Pullum, L., Cui, X., & Hinchey, M. (2012). The adaptive approach to verification of adaptive systems. In Proceedings of the fifth international c* conference on computer science and software engineering (pp. 118–122). New York, NY, USA: ACM. doi:10.1145/2347583.2347600.

The r-project. (2017). https://www.r-project.org. ([Online; accessed 20-Feb-2017]).
Sargent, R. G. (2011). Verification and validation of simulation models. In Proceedings of the winter simulation conference (pp. 183–198). Winter Simulation Conference.
Soar. (2017). http://soar.eecs.umich.edu. ([Online; accessed 20-Feb-2017]).
Statnet. (2017). https://statnet.csde.washington.edu/. ([Online; accessed 20-Feb-2017]).
Szabo, C., & Teo, Y. M. (2015). Formalization of weak emergence in multiagent systems. ACM Transactions on Modeling and Computer Simulation 26 (1), 6:1–6:25. doi:10.1145/2815502.
Tolk, A. (2015). The next generation of modeling &; simulation: Integrating big data and deep learning. In Proceedings of the conference on summer computer simulation (pp. 1–8). SCS.
Tolk, A., Diallo, S. Y., Padilla, J. J., & Herencia-Zapana, H. (2013). Reference modelling in support of M&S: foundations and applications. Journal of Simulation, 7 (2), 69–82. doi:10.1057/jos.2013.3.
Twouse. (2017). http://semanticweb.org/wiki/TwoUseToolkit.html. ([Online; accessed 20-Feb-2017]).
Vangheluwe, H. (2000). DEVS as a common denominator for multi-formalism hybrid systems modelling. In Cacsd. conference proceedings. IEEE international symposium on computer-aided control system design (cat.no.00th8537) (p. 129–134). doi:10.1109/CACSD.2000.900199.
Vangheluwe, H., & Lara, J. (2003). Foundations of multi-paradigm modeling and simulation: Computer automated multi-paradigm modelling: Metamodelling and graph transformation. In Proceedings of the 35th conference on winter simulation: Driving innovation (pp. 595–603). Winter Simulation Conference.
Wang, X., Choi, T. M., Liu, H., & Yue, X. (2016, Nov). Novel ant colony optimization methods for simplifying solution construction in vehicle routing problems. IEEE Transactions on Intelligent Transportation Systems, 17 (11), 3132–3141. doi:10.1109/TITS.2016.2542264.
Wolf, T., & Holvoet, T. (2005). Emergence versus self-organization: Different concepts but promising when combined. Lecture Notes in Artificial Intelligence, 3464, 1–15.
Yilmaz, L. (2006). Validation and verification of social processes within agent based computational organization models. Computational & Mathematical Organization Theory, 12 (4), 283–312. Retrieved from http://dx.doi.org/10.1007/s10588-006-8873.
Zeigler, B. P., Praehofer, H., & Kim, T. G. (2000). Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems (2nd ed.). Academic Press.
Zhang, Y., w. Gong, D., & Cheng, J. (2017, Jan). Multi-objective particle swarm optimization approach for cost-based feature selection in classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 14 (1), 64–75. doi:10.1109/TCBB.2015.2476796.
Zheng, Y. J., Zhang, M. X., Ling, H. F., & Chen, S. Y. (2015, Feb). Emergency railway transportation planning using a hyper-heuristic approach. IEEE Transactions on Intelligent Transportation Systems, 16 (1), 321–329. doi:10.1109/TITS.2014.2331239.

Author Biographies

Saurabh Mittal is Lead Systems Engineer/Scientist at MITRE, founder and President of Dunip Technologies, LLC and Vice President at Memberships for Society of Computer Simulation (SCS) International. He is also currently serving as Associate Editor-in-Chief for Transactions of SCS, and Editor-in-Chief of Enterprise Architecture Body of Knowledge (EABOK) Consortium. Previously, he was a fulltime scientist and architect at National Renewable Energy Laboratory, Department of Energy at Golden, Colorado, where he contributed to complex energy systems co-simulation environments. He also worked at L3 Link Simulation and Training as a contractor to US Air Force Research Lab at Wright-Patterson Air Force Base, Ohio, where he integrated artificial agents and various cognitive architectures in Live, Virtual, and Constructive (LVC) environments using formal systems theoretical model-based engineering approaches. He was a Research Assistant Professor at the Department of Electrical and Computer Engineering at the University of Arizona. Dr. Mittal served as General Chair of Springsim'17 and SummerSim'15, Vice General Chair for SpringSim'16 and SummerSim'14, and Program Chair for SpringSim'15. He is a recipient of US Department of Defense (DoD) highest civilian contraction recognition: Golden Eagle award and SCS's Outstanding Service award. He has coauthored nearly 60 articles in various international conferences and journals, including a book titled "Netcentric System of Systems Engineering with DEVS Unified Process," CRC Press in 2013 that serve the areas of executable architectures, service-oriented distributed simulation, System of Systems engineering using Department of Defense Architecture Framework (DoDAF), multiplatform modeling, intelligence-based, complex, adaptive, and autonomous systems, and large-scale M&S integration and interoperability.

José L. Risco-Martín is Associate Professor at Complutense University of Madrid. He is head of the Department of Computer Architecture and Automation. He is currently serving as General Chair of Summer Simulation Multi-Conference 2017. Previously, he was Assistant Professor at Colegio Universitario de Segovia and Assistant Professor at C.E.S. Felipe II de Aranjuez. Dr. Risco-Martín served as Program Chair for SummerSim'15, General Chair for Summer Computer Simulation Conference 2016 and Vice General Chair for SummerSim'16. He has co-authored more than 100 articles in various international conferences and journals. His research interests focus on design methodologies for integrated systems and high-performance embedded systems, including new modeling frameworks to explore thermal management techniques for multi-processor system-on-chip, dynamic memory management, and memory hierarchy optimizations for embedded systems, Networks-on-Chip interconnection design, low-power design of embedded systems, and more generally computer-aided design in M&S of complex system, with emphasis on DEVS-based methodologies and tools.

Chapter 7
Simulation-Based Software Engineering

Oryal Tanir

Abstract Software Engineering is the application of methodical principles to the planning, design, development, testing, implementation, and maintenance of software-based systems. Each phase of the Software Design Life Cycle (SDLC) addresses a different set of problems, commencing from an abstract need with the eventual goal of producing a stable working solution. To accomplish this, many different tools and techniques may be employed, from project management planning estimators to automated code testers. However, a specific tool-simulation—has found its way into almost every phase of the SDLC. As a general-purpose technique, it can be invaluable for assessing complex multifaceted solution spaces early on during the planning and design phases in a cost-effective and timely manner without the need for physically deploying possible design alternatives. A major and often overlooked element in the design of a new complex solution is the impact on the business and information technology processes. Simulation can help assess these impacts along with any process redesign that may be required. This chapter addressed these and other applications of simulation in Software Engineering.

Keywords Domain driven design · Software engineering · Synthesis · Governance models · Holistic view · Process simulation

7.1 Introduction

Software engineering, in comparison to other engineering disciplines, is relatively young. However, the field has matured rapidly as the demands and complexities in the market have grown. Simulation, in contrast, has a longer history; however, its use in software engineering was initially limited to ad hoc models. Over time, the synergy between the two disciplines has improved significantly to the point where

O. Tanir (✉)
Office of the CIO, McGill University, Montreal, Canada
e-mail: oryaltanir@gmail.com

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_7

simulation is applicable in a systematic fashion and can benefit the software engineering processes in many ways.

This section will present a brief overview of software engineering and the rational for applying simulation within the disciple. A holistic view of software engineering and application of simulation will describe the key uses of simulation. A model-driven design approach will present another major context for simulation in software engineering. The conclusions will summarize the major concepts and provide insight into some of the future directions of simulation and software engineering.

7.2 Software Engineering

Software engineering is a broad discipline, which spans and borrows practices from multiple areas such as Computer Science, Engineering, Management and Behavioral Sciences. Many definitions help qualify what software engineering is, however for the purposes of this chapter we will use the definition:

"Software engineering is that part of systems engineering that deals with the systematic development, evaluation, and maintenance of software" (Endres and Rombach 2003).

Intuitively we are dealing with a complex function, which (1) takes as its input processes, technology and resources, and (2) either optimizes or outputs a "good-enough" product that is (3) based upon market constraints (time, cost, quality). Hence, there are many layers and concerns related to software engineering—and much research has helped evolve the practice.

Some of the major concepts that are presented in this chapter and are dominant within a software engineering activity are briefly defined below.

7.2.1 The Software Design Life cycle

A key concern for any company is the choice of the software design life cycle (SDLC) model they will utilize. The SDLC describes how the software is planned, designed, implemented, and maintained. Numerous SDLCs exists such as waterfall, iterative design, spiral, agile and others. Based on the particular SDLC that is under consideration, the use of simulation will vary as it adheres to specific parts of processes that are part of the SDLC.

7.2.2 Governance Frameworks

Many software engineering activities rely upon established frameworks to ensure proper governance and oversight of activities. For example, ITIL provides a set of practices for Information Technology Service Management (ITSM)—which is broadly applicable for any software engineering organization. Other frameworks exist for particular industry domains.

7.2.3 Roles

Software engineering is not a purely technology-oriented discipline (see the holistic section later). It involves many different roles to product the final product and this aspect needs careful consideration.

7.2.4 Project Management

Project management practices work in parallel to software engineering ones during the course of a project in alignment to the SDLC. Hence some of the concerns for software design are attributable to project management ones.

7.3 Rationalization of Simulation and Software Engineering

Given, the numerous and different type of challenges that can be encountered in software engineering, it is not surprising that a tool such as simulation can be leveraged to address many of these problems. However, organizations may often find themselves in the position of justifying the use of simulation (which can be a costly affair if the right skills and resources are not available) within a project. This section reviews some of the fundamental decision points to adopt simulation within the software engineering domain. Many of these arguments will be seen later in the chapter as addressed through the application of simulation.

7.3.1 The Cost of Software Defects

One argument frequently seen in industry against the use of simulation within a software project is cost. There can be a prevailing view that the use of simulation will significantly add to the cost of a project and possibly hamper the delivery time. Hence, in this scenario a project manager who would like to avoid cost overruns and complete the project in a timely manner views simulation as a roadblock. In the context of large software projects, this argument is flawed. Many studies have shown the cost effectiveness of tools such as simulation, which can uncover design flaws early on in a project. In particular, Barry Boehm conducted extensive studies on multi-industry software projects to understand when they failed (Boehm 1981). Many other studies confirmed the fundamental finding of the research, which has since been termed as “Boehm’s law”:

> “Errors are most frequent during the requirements and design activities and are the more expensive the later they are removed.”

Although intuitive, the consequences of this statement are illustrated in Fig. 7.1. The figure’s basis is data gathered from typical software development projects and depicts Boehm’s law in action.

The graph, constructed from empirical data, shows that the cost of fixing errors later in the software cycle (i.e., in the maintenance phase) as compared to early on (i.e., in specification) is several magnitudes costlier. The magnitude of the cost can vary dependent upon the type of software, available skill-sets, organizational maturity and many other factors. Different studies have shown that the cost impact can vary from a linear to exponential one (Hait 2003; Boehm and Papaccio 1988). In addition, another problem arises when defects are uncovered later in the project phase: the may not have enough left in its budget to fix the unforeseen defects (Saultz 1997).

The important idea here is that it is prudent and cost-effective to resolve design issues as early on as possible. Software code test tools help address issues related to coding, but tools such as simulation, which utilize concepts that are more abstract, can address design level issues early on.

The arguments presented above are not always sufficient to justify the use of simulation for a given project. There are cases where simpler and more cost-effective approaches are economically or practically more feasible. For example, in some cases, mathematical models or heuristics (both of which can be leveraged through commonly used tools such as spreadsheets), may be applicable. Mathematical models usually require some basic assumptions or constraints to be in

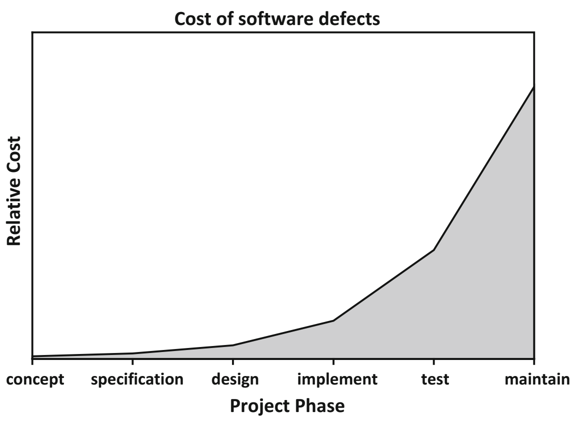


Fig. 7.1 Illustration of Boehm’s law

place before they are applicable to the problem. Such models can be applied to solve issues around contained, less complex and small software systems in a timelier manner than simulation. There is however, another reason one may not use simulation: the organization may not be able to obtain the necessary specialized resources to create, simulate, and analyze the results. There may be many reasons for this. For example, the organization may not have the funds or budget to support the work or the skill-set may not be available in the regional market. These are all factors that can influence the use of simulation in a software engineering project.

7.3.2 Business Impacts

A frequently overlooked aspect in the engineering of software is the impact on the business or end user. Software is designed against a set of requirements, however most of the time the impact on the existing processes is under emphasized or overlooked which incurs additional costs to the project due to the underestimation of the cost of change. Areas particularly vulnerable are

- processes (undocumented modification or replacement of existing processes),
- operations (miscalculation of the type of resources needed to manage the new software),
- training (miscalculation of effort needed to use software),
- and supply chain management (inadequate understanding of the full end-end integration of the software with the business).

7.3.3 Project Planning

In many cases, projects are not undertaken in isolation. The project may be part of a program composed of many other projects and understanding the dependencies between projects and the technologies that are impacted can become an exasperating problem. In addition, the timely scheduling of resources for different phases of projects is crucial for a successful and cost-effective endeavor. Simulation can be utilized in this case to evaluate different scenarios and roadmaps.

7.3.4 Time to Market

The application of simulation can also help understand the impact of time to market of the software product. In many industries, a certain window of opportunity for a software release exists, after which expected returns from the market begin to diminish. Simulation can provide indispensable insight to identify these opportunities and the various scenarios, which are both favorable and unfavorable to a software release.

7.4 Holistic View

At its core, software engineering is the business of creating a software product. This is not a simple or straightforward task for a typical project. It involves collaboration and cooperation between many different stakeholders. To accomplish the task, frameworks have been introduced to ensure the best practices and required functions are engaged at the right times. For example, ITIL provides a set of practices for ITSM, and TOGAF a framework for architecture practices (TOGAF 2005).

However, software engineering concerns are beyond just the technology of the final product. There are distinctive issues related to the business, technology, people, and governance as shown in Fig. 7.2, which play a role in a successful software practice. In each case, the application of simulation can help to ensure a successful design. This section will examine the role simulation plays within these broad categories.

7.4.1 Business Concerns

Business concerns encompass financial- and process-related issues. Within the scope of finance there are business case (expected value of the software) and budgetary (total cost of the software) issues. In contrast, process engineering has broader implications for simulation.

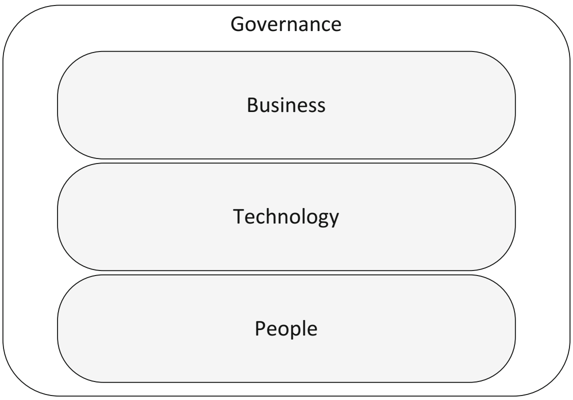


Fig. 7.2 Software engineering concerns

7.4.1.1 Financial Models

Within software engineering, financial models justify the direction of the design and viability of the project. Hence, they are typically applicable in the early planning stages of a software's life cycle. A frequently used tool is a Monte Carlo based risk simulator. Users can gain insight on the impact of design decisions and leverage the quantitative analysis for decision-making to reduce the overall risk.

For example, Monte Carlo simulation can generate statistical outcomes for specific design choices for a project and identify the respective cost implications. Then, the decision makers can make more educated project choices with an understanding of the risk that they will incur.

Another application is the use of such simulators to justify the business case for a software project. In this example, possible outcomes are the expected returns on the investment (revenue, customer satisfaction measures, and market share) to support the decision of undertaking the design commitment.

The applicability of financial simulations is mostly early in the design life cycle where access to precise data may be problematic; hence, their usefulness is dependent upon the accuracy of the available information for the models. If the margin of error of the input is high, the effectiveness of the simulated results will be questionable. However, if the model is maintainable and augmentable with new information, it can also be more useful in latter stages of a project to answer similar financial concerns.

7.4.1.2 Process Simulation

Process modeling and the subsequent activity of process simulation is a mature and common practice in software engineering with a substantial history of research and successful applications (Zhang et al. 2011; Bai et al. 2011). This is due to the fact that a complex software engineering endeavor will typically affect a multitude of business process(es). For example, new software may require a different level of interaction than its predecessor, or if a manual process will undergo some form of automation, the executable sequence of tasks may require modification (to adapt to the automated workflow). It is often difficult to make good design decisions around the process that accompanies a software product. It is also difficult to persuade the stakeholders of the benefits of a process change. These are areas where technicians successfully leverage simulation.

Process simulation is a generic enough practice that any general-purpose simulation language can be utilized. However, there is standard notation available too. For example, the Business Process Modeling Notation (BPMN) provides a standard representation for processes (White 2008). Many tools that support modeling using BPMN also have some form of simulation support. The advantage of using a standard enables the models to be more widely consumable and understandable by the business stakeholders. Hence, it can be an effective communication tool when persuading them of the benefits of a process design. They are intuitive and require knowledge of BPMN—which is not difficult to acquire in the business analysis market. However, BPMN tools use simulation in a limited fashion—mostly supporting visualization of flow in a process—that limits the potential benefits that are usually available in simulators that are more sophisticated. The typical strengths of general-purpose simulation environments such as powerful animation (visualization), statistical analysis and support of experimentation, and a rich set of random distribution support is rarely available in a BPMN tool. The drawback of such simulation environments however, is the need for specialized resources and training to use the environments, which may be beyond the means of restricted budgets. This tradeoff needs careful consideration when deciding upon the appropriate toolset to use.

7.4.2 Technology Concerns

Technological concerns relate to how simulation supports users develop the technical product during a design and implementation phase of a project. The tools common to this phase have similar characteristics to the target implementation. Similar to the business concerns, standards do prevail in many cases.

For example, UML (Unified Modeling Language) is widely adopted to model many software systems—especially in the object-oriented space (Fowler and Scott 2001). UML borrows many concepts prevalent in other areas of computer science and is amicable to the use of simulation. An example of this is behavioral simulation; where the interactions of various software elements can undergo simulation before physical code creation or test. In such cases, simulation identifies important design constraints such as

- The dependencies between different software components based upon their interactions. Discrete-event simulation can execute the flow between software objects to determine bottlenecks based upon the message flowing between each.
- Detection of redundancies is possible but often difficult. Simulation models the flow and specialized test tools detect patterns, which may indicate duplicate behavior in the design.
- Reachability analysis of behavioral constructs such as methods by simulation or simulation-based petri-nets. In such cases all-possible execution paths of the code is simulated which permits detection of code that is not reachable (dead code) or other problematic code behavior such as infinite loops.

To enable these types of analyses, simulators work in close conjunction with the design tools. For example, if the design utilizes UML as a model notation, tools which support this, may incorporate (among others) state-chart simulators. Using a state based simulator, all possible interactions between states can be exhaustively generated and virtually tested to ensure that the overall software behavior is compliant to its specifications.

When software is to be part of an embedded system, it is cost-effective to simulate its functionality and physical characteristics (such as performance and latency) before the fabrication process. Simulation is a technique that is an integral part of the toolsets used in this process as well [such as VHDL or Verilog simulators] (Navabi 2007).

Another standard that is usable at this stage is the Business Process Executionable Language BPEL. BPEL foundation is XML and web services and is a process oriented executable language, primarily used in web based integration and design. It fits well in organizations that have adopted a Service-Oriented Architecture (SOA) approach to software design and integration. Major software vendors support BPEL and consequently their tools have BPEL simulators to aid in the design of web services. A major benefit of simulation at this stage is the ability to simulate orchestration or choreography

- Orchestration of web services implies the use of a central web service, which systematically requests services from other web services and then generates some sort of output or result.
- Choreography of web services does not require a central control. In contrast, each service accepts and sends messages to a limited set of services—their combined interaction or "choreography" results in the desired overall behavior.

Constructing and integrating web services in the above manner becomes a difficult task as the number of webs services increase and the dependencies between each and existing services becomes difficult to manage. The BPEL process models define these interactions—since their interactions can readily be described in terms of processes. Once the BPEL model creation is complete, essentially describing the overall behavior, the simulation activities will:

- Validate the flow of information between web services, which the BPEL processes depict. This validation ensures that the logical flow within each service contributes the correct sequence of activities that will generate the overall process behavior.
- Create an impact analysis of all existing and proposed web services. Rather that detecting interaction problems between services in the field, simulation of their BPEL counterparts permits design engineers to detect potential design issues in the lab. The simulation activities can uncover
  - performance impacts (some service may be a bottleneck) due to too many dependencies or poorly designed logical flow in the service,
  - logical flaws (a process is not flowing as expected),
  - reliability concerns (the failure of certain services may be critical to many other system processes),
  - and inefficient services (too many calls and time spent on a service may indicate it needs to be decomposed to simpler ones).

7.4.3 People Concerns

People concerns are part of the project management role; however, they affect the success of a software engineering project significantly. In particular, planning and estimating the capacity to perform work at different stages in the software life cycle can be difficult. The main tool used by managers is still spreadsheets and Gantt charts; however, simulation has made headway in capacity planning and estimation.

General-purpose simulation tools are dominant in this practice space. Some vendors do provide a combination of project management and simulation capabilities; however, the simulation features are mostly cosmetic or primitive. The functionality can improve in the future as the market for such features increase.

Current general-purpose simulation techniques use a combination of process simulation and resource optimization scenarios. Main use cases for this are

- Decision-making for multiple project and resource trade-offs: There can be cases where different planned projects compete for the same resources at different time lines. Optimization or near-optimization of timelines and resources across multiple projects is a difficult task which simulation is well suited to perform. The constraint is that simulations in this area require specialized skill-sets, which may be too costly or difficult to acquire.
- Capacity planning within a project: Simulation can produce capacity scenarios from common constrains such as hourly wages, scheduling rules, resource availability, skill-set modifiers, and task times (based on heuristics or statistical data).
- Automation versus manual work: In many cases there may be concerns with the trade-offs between automation of certain tasks or managing them manually. Sometimes referred to "people versus technology" scenarios, simulation can provide insight on cost and time implications.

7.4.4 Governance

Governance, as it relates to simulation use in software engineering, is often an oversight. Some simulation activities may be dispensable and used only for a particular decision point in the overall software design and never again. However, in more mature software teams many of the knowledge acquired through simulation is retained or reused. BPEL-driven simulation is one example where a rich knowledgebase of models develops over time. The data and accuracy of the simulations become better with the addition of new models. Governance of the models becomes a necessary part of the software design life cycle. Some common concerns to address are

- Which role is accountable for the model? In cases where the simulation is ad hoc, then it may be the simulationist or their manager. However, for environments that reuse simulation objects, then a role is required to ensure model compliance with the organization's rules and principles. This is usually part of an Enterprise Architecture role, but can also be the accountability of a technology or service manager.
- Which role approves the model? A governance process is required to ensure that the principles related to the design and output of the model are valid. This is typically different from the accountable resource above.
- Terms of engagement of simulation or the conditions when simulation is used and its expected outcomes. It is very important that the expectations of a simulation exercise are the same for all the stakeholders. If no clear definition and boundaries are set for the simulation activity, it can grow or not meet expectations.

The above are typical of software engineering governance issues and it makes sense to utilize similar governance practices for the simulation practice within the software engineering framework. Some important artifacts that need to be defined are

- Model design principles: These are basic principles to abide by when constructing models. They will represent the adopted notation and basic guidelines as well as design patterns when constructing a simulation.
- Data principles: These are basic rules to ensure that data that is used in simulation models is vetted properly and it is handled and interpreted correctly by the modeler. Rules for selecting the correct random distribution which fits a given data set is an example of this.
- Simulation execution principles: The rules determining how long to run simulations and iterations of experiments are common concerns.
- Requirements for a simulation exercise: There can also be guidelines for establishing is a certain project is suitable to undergo a simulation exercise. Concerns such as the quality of the data, complexity of the system under consideration and correct level of expectations are typically addressed.
- Roles and responsibility charts: The expected resources and responsibilities for the simulation project needs to be defined and allocated.
- Decision-making principles or process: The decision-making principles or "rules of engagement" solidify when key decisions resulting from simulation outcomes are made.
- Escalation procedures: A mechanism needs to be in place to resolve issues quickly (i.e., lack of data, resources, time).

As part of the governance, the validity of the simulation studies also needs close examination and understanding. Studies in various industrial fields have shown that the validity of simulation results within a software engineering activity can be quantified and its risks mitigated (França and Travassos 2015). Hence the body of research in this area can help support practitioners successfully compete their simulation ventures.

7.5 Domain-Driven Software Design

Domain-driven software design is a model-based approach in software engineering and simulation is an intricate component at different stages of the methodology. The typical previously mentioned software engineering concerns are applicable here as well. However, the model-driven approach lends itself to model reuse (which permits the creation of a more permanent simulation expertise) and subsequently the application of simulation in this case is methodological and systematic.

There are variations of the approach, but the high-level elements (not all of which need to be used—depending upon the methodology) are illustrated in Fig. 7.3.

The major components are

- **Domain model**: The starting point is the capture of abstract ideas in a domain model. Such models provide a domain specific language or notation suitable to represent the structure and behavior of the design concepts that are of importance (Erdogmus and Tanir 2002). The model must be familiar and versatile enough so that the user captures and validates ideas quickly before undertaking any detailed design decisions. Representations at this level are the high-level specifications for the conceptual system. Successful environments will have strong visual user interfaces to improve productivity and reduce the learning curve. When using simulation, the selection of the appropriate domain model is important. As a requirement, the model needs to support the simulation language or formalism it will be utilizing.
- **Model Checker**: A domain model often relies on a model checker. Such functionality is composed of established formal verification techniques to ensure a sound basis for the model that is developed. A formal verification ensures that the downstream steps to follow will be less prone to errors due to inconsistent representation of model elements.

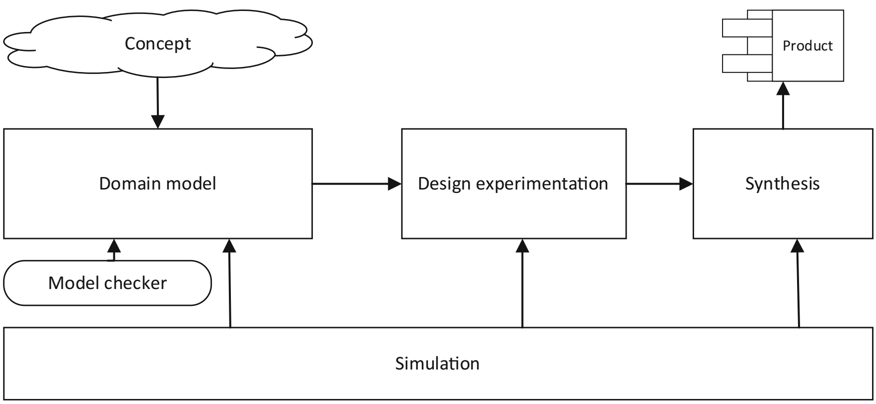


Fig. 7.3 Domain-driven software design components

While model checkers use formal verification techniques, some may employ simulation when formal techniques are not feasible. This can occur in cases where the domain model uses a nonformal language and cannot be validated using such techniques (Tanir et al. 1996). This is often the case when the language supports very abstract concepts to allow a versatile user experience. In such instances, model objects may not bind to a specific behavior or structure early in the design, but missing details populate and complete the missing pieces as the design progresses.

Simulation can bridge the conceptual gap by using semi-formal notations to provide statistics-founded analysis. For example, high-level, colored, or statistical petri-net based formalisms (Haas 2002) are often applicable using a combination of traceability analysis and token simulations to validate the structure or behavior of the model (Tanir and Erdogmus 1999).

- **Design Experimentation**: Experimentation can imply many types of activities. However, it encapsulates those that permit
  - Design space exploration: Tools to support analysis of alternative designs.
  - What if analysis: Such tools will let users change parameters and structure and compare outputs or outcomes.
  - Scenario decision-making tools: Statistical support and directed scenario analysis leverage mathematical techniques to validate scenario outcomes to guide decisions to optimal or near-optimal design choices (Miranda 2002).

Many of the tools employed at this stage will utilize simulation as the principle method of execution of models and comparison of multiple designs. The tools are either specialized or general-purpose and depend upon the design space that is under consideration.

- **Synthesis**: The eventual objective of the modeling approach is to produce executable code from high-level specifications. Synthesis tools accomplish this task. The high-level model that has been verified and validated through model checkers and simulation can now be "synthesized" into code. Synthesizers are technology specific and require a design represented in a formal specification language particular to a domain. For example, Java code synthesizers may require a UML (Universal Modeling Language) based model, whereas an embedded application will use a synthesizable VHDL (Very high-speed integrated circuit Hardware Description Language) model.

The synthesis stage requires a set of different technologies to accomplish the tasks. While most of these are beyond the scope of this chapter, simulation based tools are often part of the synthesis package (Tanir 1997). Synthesis implies a transition from a semi-formal notation that is prevalent during design experimentation to a structured standard one for the target code. The latter is generally not based on a formal representation, but more on a standard notation. Hence, simulation validates the resulting "synthesized" target model.

There are also new simulation concerns at this stage. For example, simulating latency, cycle times, and probability of failures are part of the validation practices. The notation in the simulators in this case are close to the target formalism (i.e., VHDL or UML) and therefor part of a synthesis software offering. BPEL, which was introduced earlier, can also be part of the simulation and synthesis activities that are related to the design of web services.

- **Software Package**: The Software package is the resulting product. At the minimal it is executable software code, but it will typically also contain supporting software by products such as
  - Automated test cases that validate the functionality of the code against the initial (domain language) specifications or business requirements.
  - Service level agreements that may be part of the requirements of the software.
  - Self-test code can be included for systems that will be synthesiz-able to silicon. This will permit the testing of fabricated compo-nents in a non-intrusive manner.
  - Software documentation that describes the functionality of the code and any changed components (if a history exists).

As can be seen, the use of simulation within software engineering is quite open and applicable across a broad range of activities and domains.

7.6 Conclusion

Simulation use in software engineering has progressed from ad hoc throwaway models to reusable ones. This trend will further improve in the future as more vendors adopt or improve their simulation offerings. General-purpose simulators will still prevail in many software engineering activities since specialized simulators do not meet all the needs across a software design life cycle. Many of the concepts developed in the artificial intelligence domain is now technically and financially feasible to be applied in certain circumstances to utilize simulation and design project in new and novel ways (Elzas et al. 1989). For example, models could potentially adapt to proposed design changes based on design patterns and best practices to aid the experimentation process.

As with any software tool, standardization of the use of simulation tools and the way in which they are integrated into the software engineering processes is important for the success of any software project.

Review Questions

1. What are some strong arguments that can be made to bolster a business case for adopting simulation within a software engineering project?
2. Under what circumstances would simulation not be a feasible choice within the context of software engineering?
3. Define and elaborate upon the basic elements of a holistic view of software engineering.
4. Which area of software engineering has simulation played the most prevalent role and considered a mature practice?
5. What are key deliverables that should be part of a governance process for simulation?

References

Bai, X., Zhang, H., & Huang, L. (2011). Empirical Research in Software Process Modeling: A Systematic Literature Review. 2011 International Symposium on Empirical Software Engineering and Measurement. doi:10.1109/esem.2011.43.

Boehm, B. W. (1981). Software engineering economics. Englewood Cliffs, NJ: Prentice-Hall.

Boehm, Barry W. & Papaccio, Philip N. "Understanding and Controlling Software Costs," IEEE Transactions on Software Engineering 14, 10 (October 1988): 1462–1477.

Elzas, M. S., Ören, T. I., & Zeigler, B. P. (1989). Modelling and simulation methodology: Knowledge systems' paradigms. Amsterdam: North-Holland.

Endres, A., & Rombach, H. D. (2003). *A handbook of software and systems engineering: Empirical observations, laws, and theories*. Harlow, England: Pearson Addison Wesley.

Erdogmus, H., & Tanir, O. (2002). Advances in software engineering: comprehension, evaluation, and evolution. New York: Springer.

Fowler, M., & Scott, K. (2001). UML. Paris: Campus Press.

Haas, P. J. (2002). Colored Stochastic Petri Nets. Stochastic Petri Nets, 385–445. doi:10.1007/0-387-21552-2_9.

Hait, C. (2003). Economic impacts of inadequate infrastructure for software testing: final. Place of publication not identified: Diane Pub Co.

França, B. B., & Travassos, G. H. (2015). Simulation Based Studies in Software Engineering: A Matter of Validity. CLEI electronic journal, 18(1). doi:10.19153/cleiej.18.1.4.

Miranda, E. R. (2002). Computer sound design: synthesis techniques and programming. Oxford: Focal.

Navabi, Z. (2007). VHDL: modular design and synthesis of cores and systems. New York: McGraw-Hill.

Saultz, J., & Kalathil, B. (n.d.). Rapid prototyping of application-specific signal processors (RASSP) approach to meeting 4X. Annual Reliability and Maintainability Symposium. doi:10.1109/rams.1997.570023.

Tanir, O. (1997). Modeling complex computer and communication systems: a domain-oriented design framework. New York: McGraw-Hill.

Tanir, O., Agarwal, V., & Bhatt, P. (1995). A specification-driven architectural design environment. Computer, 28(6), 26–35. doi:10.1109/2.386983.

Tanir, O., Agarwal, V. K., & Bhatt, P. C. (1996). DASE: An environment for system level telecommunication design exploration and modelling. Computer Aided Systems Theory—CAST '94 Lecture Notes in Computer Science, 302–318. doi:10.1007/3-540-61478-8_85.

Tanir, O., & Erdogmus, H. (1999). A Framework for Topological Re-use. In M. Fayad, D. C. Schmidt, & R. E. Johnson (Authors), Implementing application frameworks: object-oriented frameworks at work. New York: Wiley.

TOGAF: version 8.1, Enterprise Edition. (2005). San Francisco: Open Group.

White, S. A., & M. (2008). BPMN: modeling and reference guide. Florida: Future Strategies, Incorporated.

Zhang, H., Jeffery, R., Houston, D., Huang, L., & Zhu, L. (2011). Impact of process simulation on software practice. Proceeding of the 33rd international conference on Software engineering - ICSE '11. doi:10.1145/1985793.1985993.

Author Biography

Oryal Tanir is the Enterprise Architect in the Office of the CIO and professor at McGill University. He has been involved in simulation since 1984 and has practiced both as a practitioner and as a researcher in software engineering since 1986. He has worked in different industries such as telecommunications, travel, and higher education with experience in many facets of IT such as; Simulation, Software Design, Enterprise Architecture, System engineering, and Business Process Management. In his current capacity at McGill University, Dr. Tanir influences the strategies and directions of software engineering within the Higher Education area. His research interests have been concentrated in various topics in simulation and the automated synthesis of complex systems. He has numerous publications in software engineering and simulation and has lead or collaborated in work in this area with McGill University, University of Toronto, Ottawa University, and University of Waterloo. He is also the author of the novel "The Falcon's Arrow." He frequently contributes to the Canadian Research granting boards in different capacities.

Chapter 8
Simulation-Based Architectural Design

Rhys Goldstein and Azam Khan

Abstract In recent decades, architects have turned to computer simulation with the hope of designing more functional, sustainable, and compelling buildings. In such efforts, it is important to regard buildings not merely as static structures, but rather as complex dynamic systems driven by highly stochastic elements including the weather and human behavior. In this chapter, we describe how simulation has impacted architectural design research and practice. A multitude of simulation tools have been developed to model specific aspects of a building such as thermodynamics, daylight, plug loads, crowd behavior, and structural integrity under internal and external loads. Yet numerous challenges remain. For example, although many factors influencing buildings are interdependent, they are often analyzed in isolation due to the development cost associated with integrating solvers. A systems approach combining visual programming with state-of-the-art modeling and simulation techniques may help architects and building scientists combine their expertise to produce integrated complex systems models supporting emerging paradigms such as generative design.

Keywords Architecture · Building simulation · Building science · Energy modeling · Sustainability · Systems approach · Discrete event simulation · Design tools · Computer-aided design · Building information model · Heat transfer · Occupant behavior · Daylight simulation · Co-simulation · Model integration · Visual programming · Dataflow programming · Parametric design · Generative design · Performance metrics

---

R. Goldstein (✉) · A. Khan
Autodesk Research, 210 King Street East, Suite 500, M5A 1J7 Toronto, ON, Canada
e-mail: rhys.goldstein@autodesk.com

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_8

8.1 Introduction

In Chap. 1, Ören et al. provide a number of reasons why simulation is used in general, and many of these reasons apply to the design and optimization of functional, cost-efficient, safe, healthy, sustainable, and visually compelling buildings. Simulation is often used when the real system does not exist, which is necessarily the case when a new building is designed. Simulation is also used when the real system is too slow; thermal performance and daylighting require at least year to properly observe, which is inconveniently long when designing a retrofit for an existing building. Simulation is used when physical experiments are dangerous, unacceptable, or costly, all of which dictate that we should not wait for a building to collapse before simulating its structural integrity under internal and external loads. Finally, simulation is used when the variables of a system cannot be controlled. Two significant, highly stochastic variables influencing the performance of a building are the people who occupy it and the weather. Neither human behavior nor the weather can reasonably be controlled for experimentation purposes, yet a wide range of behavioral patterns and environmental conditions can be tested in a virtual setting.

The physical complexity of a building is evident by simply looking at a *building information model* (BIM) such the one shown in Fig. 8.1. These models, which now enjoy widespread use in the architecture, engineering, and construction

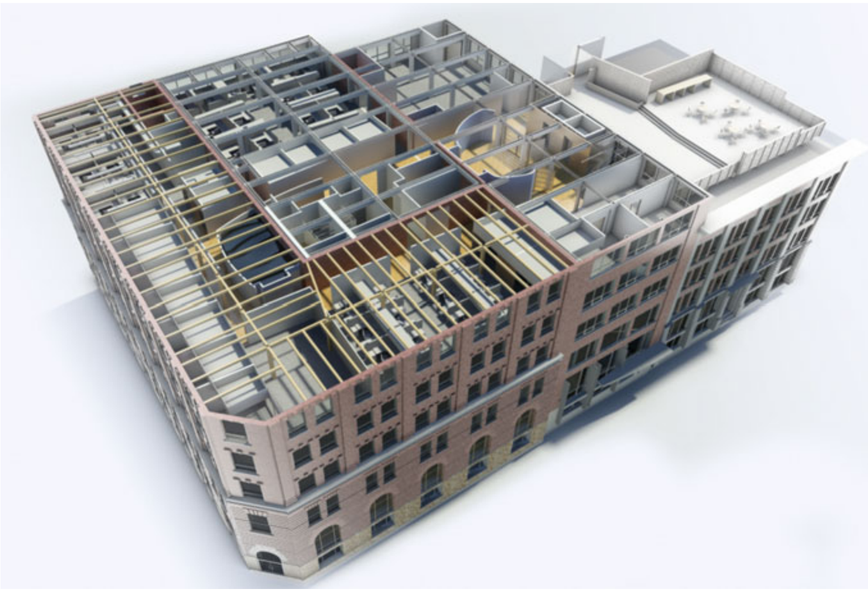


Fig. 8.1 A building information model (BIM) representing the 210 King Street East heritage building in Toronto, Canada

(AEC) industry, combine core building elements such as walls, slabs, windows, and doors with more detailed elements such as furniture, lighting fixtures, and heating, air-conditioning, and ventilation (HVAC) components. By incorporating properties such as materials and room types, BIMs have the potential to supply much of the static information required for highly detailed building simulations. However, buildings should be regarded not as static structures containing physical objects, but rather as dynamic systems involving numerous interacting forces and active entities such as the outdoor climate, electrical/mechanical equipment, and human occupants. The many processes that unfold throughout a building's lifetime give it an additional level of complexity that is only partially accounted for by the simulation tools currently available to architects.

In this chapter, we review some of the most prevalent simulation tools used in building design and engineering practice (Sect. 8.2), highlight a sample of recent and ongoing research efforts in the field (Sect. 8.3), and discuss the potential role that state-of-the-art modeling and simulation (M&S) techniques might play in helping various stakeholders collaborate in the development of next-generation simulation-based design tools (Sect. 8.4). A systems approach for developing building simulation software—based on research from the M&S community—would support the integration of both existing and future models of building thermodynamics, lighting, and occupant behavior. It would also ease the exploration of emerging design paradigms such as those involving the automatic generation of design options, referred to as *generative design*.

Buildings have a tremendous impact on the natural environment, accounting for 41% of all energy consumption and 72% of electricity use in the United States (Livingston et al. 2014). Moreover, they have a less quantifiable but equally significant effect on human experience, as in today's society people spend much of their time in and around buildings. Decision support for building design is therefore one of the most potentially beneficial of all uses of simulation.

8.2 Current Simulation Tools for Architecture

A wide variety of building simulation tools exist for assessing various aspects of buildings. Focusing first on energy-related software, there are 147 tools currently listed in the Building Energy Software Tools Directory (BEST-D). Many of these tools, however, are based on a few core simulators, such as Radiance (Ward 1994) for lighting and EnergyPlus (Crawley et al. 2001), DOE-2 (Curtis et al. 1984), or ESP-r (Aasem et al. 1994) for whole building energy simulation.

To perform an analysis using a detailed BIM and one of the whole building energy simulation tools, the BIM must first be converted into an energy analytical model. In this highly simplified type of model, buildings are represented as networks of polyhedral spaces, each assumed to have a uniform temperature (Clarke 2001). Large rooms such as corridors or atria can be converted into several adjacent spaces separated by arbitrary boundaries, allowing temperature to vary in steps within the indoor area. Surface elements of various materials and thicknesses resist the flow of heat among spaces separated by walls, slabs, and other physical barriers. The mathematics underlying this basic method was largely developed prior to the 1990s when the limited availability of computing power necessitated such approximations. Despite substantial increases in computing resources and decades of subsequent building simulation research, the early approximations remain in use to this day. Fortunately, the task of converting detailed architectural models into simulation-ready energy models is becoming increasingly automated. Figure 8.2 shows the spaces and surface elements of an energy model created automatically from a BIM.

Conveniences such as automated BIM-to-energy-model transformation encourage architects to incorporate technical analyses traditionally performed by engineers in later stages of the building design process. Because many of the decisions that affect the energy efficiency of a building are made by architects at the early design stage, the increased use of energy simulation by designers is seen as a promising strategy toward realizing more sustainable built environments. As emphasized by Bazjanac et al. (2011), challenges such as missing data exist in providing designers with accurate whole building energy results. Indeed, Berkeley et al. (2014) find that even professional energy modelers produce dramatically divergent estimates given the same building and the same modeling tool, highlighting a general need for future developments in building energy simulation software.

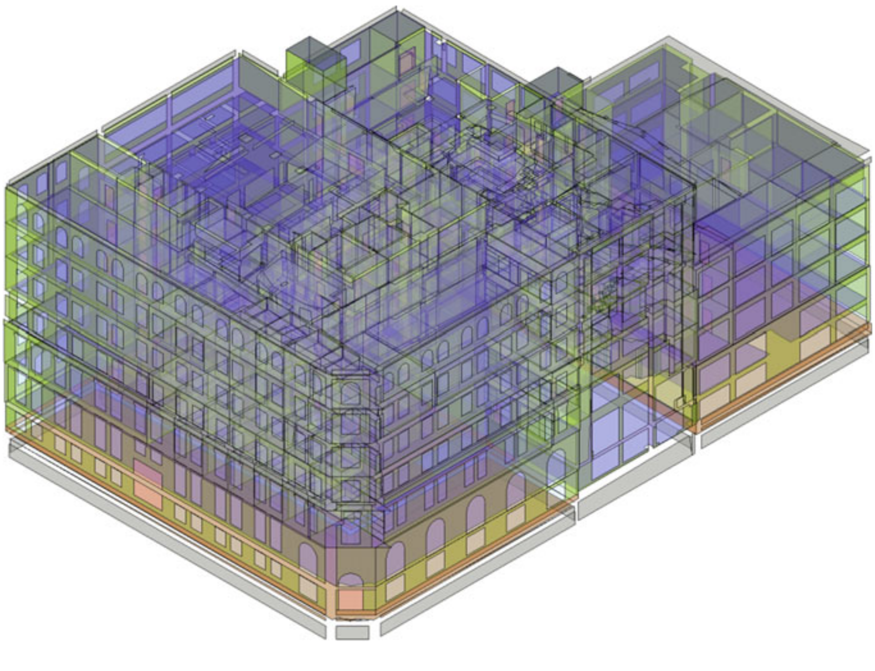


Fig. 8.2 An energy analytical model of the 210 King Street East building created automatically within the BIM-authoring tool Autodesk Revit 2016

Aside from energy-related analyses, simulation has a number of applications in building design and engineering. These include traditional uses such as structural analysis, as seen in Fig. 8.3, and more recent applications such as the multi-agent simulation of crowds for predicting issues related to pedestrian flow or building evacuation. Figure 8.4 shows a snapshot of a multi-agent simulation performed

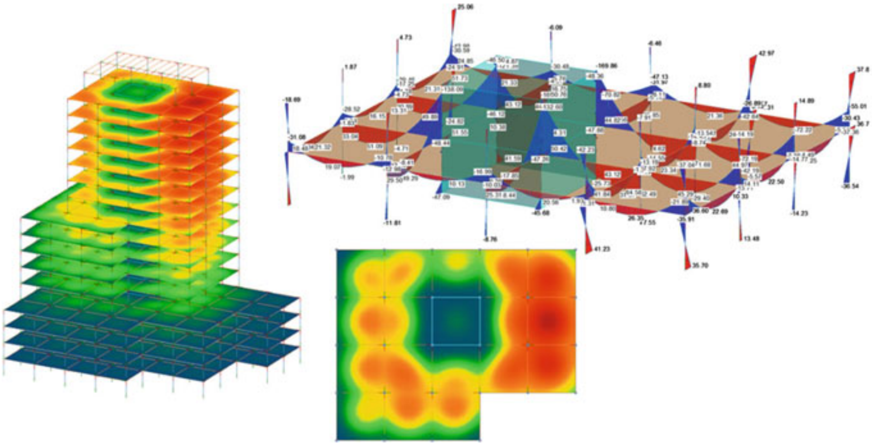


Fig. 8.3 Structural analysis results produced by the Revit BIM-authoring tool

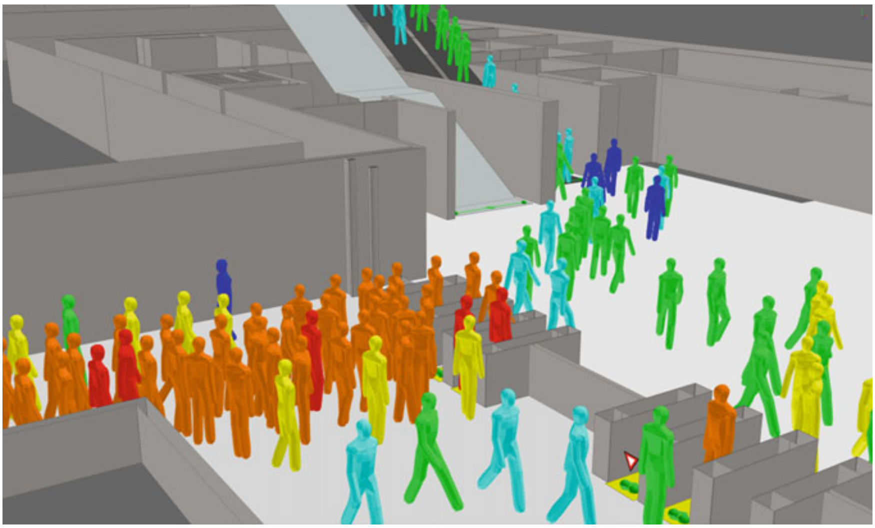


Fig. 8.4 Multi-agent crowd flow simulation performed by the MassMotion design tool (Image courtesy of Erin Morrow, Oasys/Arup)

using the commercial tool MassMotion (Morrow 2010; Morrow et al. 2014), intended for the design of transportation hubs, healthcare facilities, arenas, and other built environments where crowd behavior demands careful attention.

8.3 Architectural and Building Science Research

The large and growing body of research into simulation-based building design can be regarded as occurring within two mostly distinct communities: one primarily involving architects, the other engineers.

Architectural researchers who investigate simulation and other computational methods present their work at designer-oriented venues such as the ACADIA conferences (ACADIA: Association for Computer-Aided Design in Architecture), CAAD Futures (CAAD: Computer-Aided Architectural Design), eCAADe (Education and research in CAAD in Europe), CAADRIA (CAAD Research in Asia), Smartgeometry, and Rob|Arch (Robots in Architecture). In addition to structural and environmental performance, much attention is paid to qualitative measures such as building aesthetics and the manner in which humans perceive, experience, and respond to the built environment. In addition, researchers in this area are becoming increasingly interested in how emerging fabrication techniques, such as the use of robots, can aid the realization of historically intractable designs.

On the engineering side, research into simulation-based building design is generally referred to as building science. One of the primary goals of this research community is to optimize building performance, essentially maximizing the comfort of a building's occupants while minimizing both operational costs and the building's negative impact on the natural environment. Much of the work is presented at the regional and international conferences of the International Building Performance Simulation Association (IBPSA). An international IBPSA conference occurs every two years (recently 2013, 2015, etc.), and the regional conferences around the world are typically hosted on the alternate years. A comprehensive overview of the state of the art in this area can be found in *Building Performance Simulation for Design and Operation*, edited by Hensen and Lamberts (2012). The books' chapters provide a nearly complete list of the domains in which members of the community specialize, including weather, occupant behavior, heat transfer, ventilation, occupant comfort, acoustics, daylight, moisture, HVAC systems, micro-cogeneration, building operations, and government policy pertaining to buildings and energy.

Although most research efforts relevant to simulation-based building design tend to fall into one of the two broad but relatively distinct disciplines, it is well understood that the overarching goal of improving the built environment and making it sustainable is shared among architects and engineers, and requires collaboration among all stakeholders. Hence there are many who present their work at both the designer-oriented and engineer-oriented conferences (i.e., ACADIA and IBPSA), facilitating the exchange of ideas between communities. Since 2010 there has even been a venue—the Symposium on Simulation for Architecture and Urban Design (SimAUD)—largely dedicated to promoting discussion between designers and building scientists, with simulation tools and techniques serving as a common focus.

In the remainder of this section, we highlight a small sample of recent academic research presenting new ideas and recently developed tools that advance the use of simulation in building design. All of these works feature elements familiar to the general M&S community, including modeling languages, modern computing technology, and co-simulation.

8.3.1 Example of Occupant Behavior Research

Schaumann et al. (2015) propose a graphical modeling language for creating narratives that drive the behavior of simulated occupants in not-yet build environments. The focus is on hospital design, for which architects need to understand the complex reoccurring patterns of behavior exhibited by interacting doctors, nurses, patients, and visitors. An example of a narrative is the checking of a patient by a doctor–nurse team. By visualizing a multi-agent simulation of this routine, as shown in Fig. 8.5, a designer may gain insights into whether a particular design option promotes the efficient performance of this activity, or hinders it with an inefficient layout or with probable interruptions by hospital visitors.

Multi-agent approaches such as that of Schaumann et al. (2015) represent a radical departure from the current standard practice in whole building energy

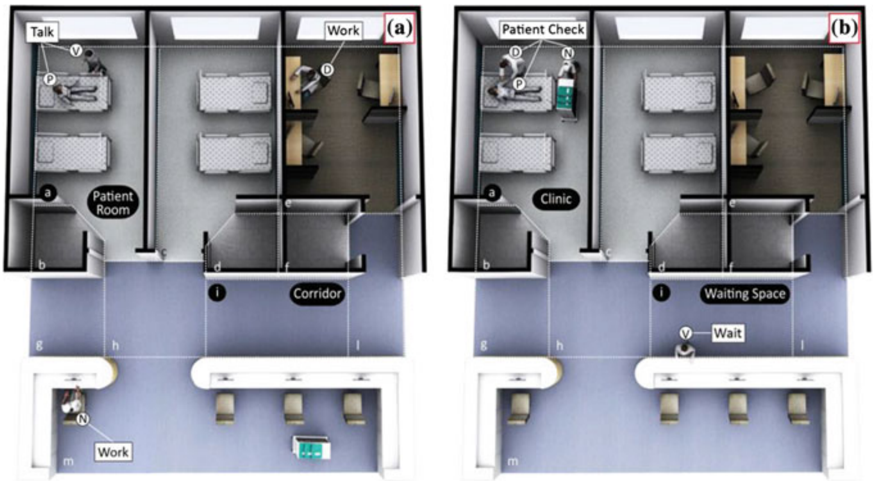


Fig. 8.5 Visualization of occupant behavior in a hospital environment. From Schaumann et al. (2015); reprinted with permission from the authors

modeling, where occupant behavior is modeled using fixed profiles. These profiles typically give aggregated hourly information about the degree to which a space, electrical appliance, or building system is used. The most prominent examples of these profiles are those found in ASHRAE (2004) and subsequent versions of Standard 90.1. Although fixed profile models enjoy widespread use, higher fidelity behavioral models would accommodate new quantitative analyses—such the evaluation of automatic lighting systems based on motion detectors—as well as qualitative investigations that would likely appeal to architects. As mentioned in Sect. 8.2, multi-agent simulation tools are available for pedestrian flow and evacuation, but less so for other normal day-to-day activities of people in buildings.

8.3.2 Example of Daylight Simulation Research

Jones and Reinhart (2015) introduce a new tool called Accelerad, which combines GPU technology with other optimization techniques to perform daylight simulation up to 24 times faster than Radiance with similar areas. Results for two indoor environments are shown in Fig. 8.6. This research takes advantage of two broad opportunities in the discipline. First, it exploits computing technology that has emerged after much of the core research on conventional energy simulation tools was conducted; that is, technology such as the GPU, developed during the 1990s or later. Second, it aims to satisfy the needs of architects, as opposed to engineers, in this case by delivering the speed necessary to gain rapid feedback and explore a greater number of design options.

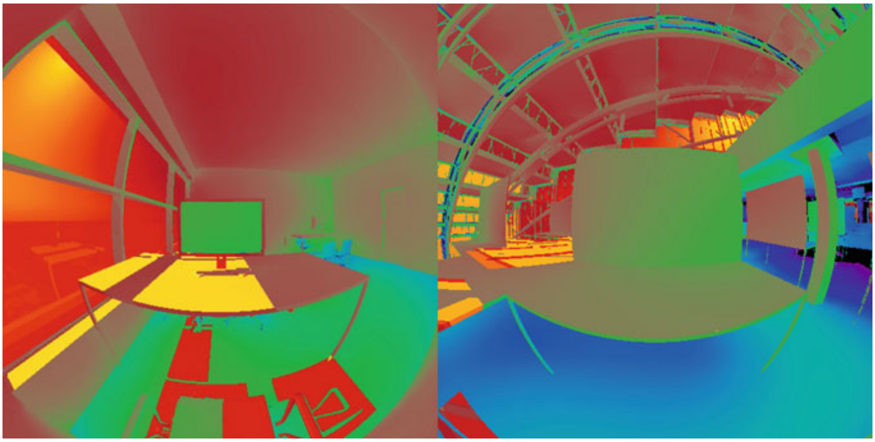


Fig. 8.6 Daylight simulations performed by the Accelerad using GPU technology. From Jones and Reinhart (2014); reprinted with permission from the authors

Table 8.1 External tools linked to or used by the Building Controls Virtual Test Bed (BCVTB) co-simulation environment

External tool	Purpose
Dymola (Modelica)	HVAC system modeling and building controls modeling
Simulink	Building controls modeling
MATLAB	Building controls modeling and data analysis
EnergyPlus	Whole building energy simulation
ESP-r	Whole building energy simulation
Radiance	Lighting simulation
TRNSYS	System simulation
BACnet stack	Data exchange with building automation systems
A/D converter stack	Data exchange with analog/digital converter
Functional Mock-up Units (FMU)	Co-simulation and model exchange

8.3.3 Example of Co-simulation Research

Wetter (2011) introduces the Building Controls Virtual Test Bed (BCVTB), a co-simulation environment linking an assortment of building simulation tools. The environment is built on the multi-paradigm modeling software Ptolemy II (Brooks et al. 2007), uses the Functional Mock-up Interface (FMI) as the standard to support co-simulation (Nouidui et al. 2013), and currently links the tools listed in Table 8.1.

Although the factors influencing buildings are often analyzed in isolation, the impressive number of tools integrated by BCVTB supports the notion that buildings are complex systems involving a variety of interacting processes. No single simulator fully accounts for thermodynamics, light propagation, weather, human behavior, and mechanical systems, and hence co-simulation is perhaps the only way to realize a truly comprehensive building performance model without rewriting a large portion of existing code. Using co-simulation, existing tools share information once per time step, or several time per time step, depending on the strategy adopted by the moderating software. The IBPSA community features several projects in which two tools are integrated via co-simulation, examples being ESP-r & Radiance (Janak 1997) and ESP-r & TRNSYS (Beausoleil-Morrison et al. 2011). The BCVTB is unique in the number of tools it connects, as well as the fact it is intended to support the incorporation of additional tools.

At present, co-simulation appears to be the most popular approach for integrating building simulation algorithms that are not currently implemented in any single tool. In the future, other integration approaches may be beneficial. Goldstein et al. (2013) demonstrate the use of a formalism-based model-independent simulator such as DesignDEVS (Goldstein et al. 2016) as a technological alternative to co-simulation. In addition, adaptive time steps and quantized state solvers are mentioned as mathematical alternatives to the numerical integration strategies which currently dominate building performance simulation research. As explored in the next section, these ideas from the M&S community have the potential to promote collaboration in the development of next-generation building simulation methods, possibly leading to more architect-friendly tools that take advantage of modern computing technology and better accommodate future design paradigms.

8.4 A Systems Approach for Simulation-based Architecture

An opportunity exists to dramatically improve collaboration among architectural researchers and building scientists. This can be done by applying state-of-the-art techniques from the M&S community, which investigates aspects of computer simulation that span disciplines. Our long-term vision is that comprehensive model-dependent simulators such as EnergyPlus and ESP-r could eventually be replaced by a repository of considerably more focused models with a common interface. These new models, contributed by members of the building simulation community, would be integrated in various combinations, and the most successful configurations could be packaged for the benefit of practitioners. A platform of this nature would promote the ongoing improvement of building simulation methods, and allow a much larger group of researchers to participate in the development process. Here we outline a collaborative systems modeling approach particularly well-suited to the discipline of architectural design. Other ideas from the M&S community also merit exploration in this application area.

The underlying principle we follow is to build upon architects' familiarity with certain programming techniques, namely conventional procedural programming and dataflow visual programming. Procedural programming, involving assignment instructions and control flow structures such as "if" statements, is currently taught to students in a wide range of fields including building science and architecture. Although a typical designer has less programming experience than a typical computer scientist or systems engineer, we can rely to some extent on widespread knowledge of basic programming concepts. Dataflow programming, by contrast, is a style of programming that has become especially popular in the architectural research community as a technique supporting *parametric design* (Woodbury 2010). As shown in Fig. 8.7, parametric design tools such as Grasshopper

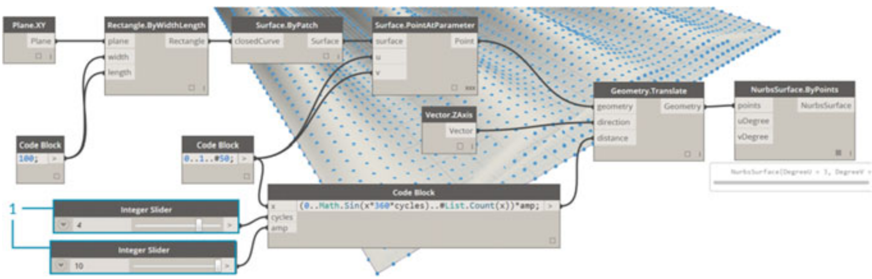


Fig. 8.7 An example of dataflow programming in dynamo

(Mode Lab 2015) and Dynamo (Autodesk 2017) allow building geometry to be defined programmatically and modified interactively in visual programming environments integrated with design tools. As observed by Doore et al. (2015) in another discipline (multimedia), the popularity of paradigms such as dataflow programming creates a favorable environment for introducing other M&S concepts.

The systems modeling approach we describe combines dataflow programming with the Discrete Event System Specification (DEVS), the latter of which is a modeling formalism generally applied using procedural code inside composable modules exhibiting a common interface (Zeigler et al. 2000). The overall approach is illustrated by a set of visual interfaces designed by Maleki et al. (2015), some of which are shown in Fig. 8.8. The dataflow elements, appearing at the left and right sides of the interface, are responsible for the initialization of a simulation as well as the aggregation of its results into performance metrics and other statistics. The DEVS elements, placed in the central column of the interface, handle the simulation itself, which captures the evolution of a real-world system over time. As is common among modeling paradigms from the M&S community, scalability is achieved in part via the use of hierarchies. The overall interface represents a *system node*, and the four central nodes within are also system nodes that potentially encapsulate their own dataflow and DEVS elements.

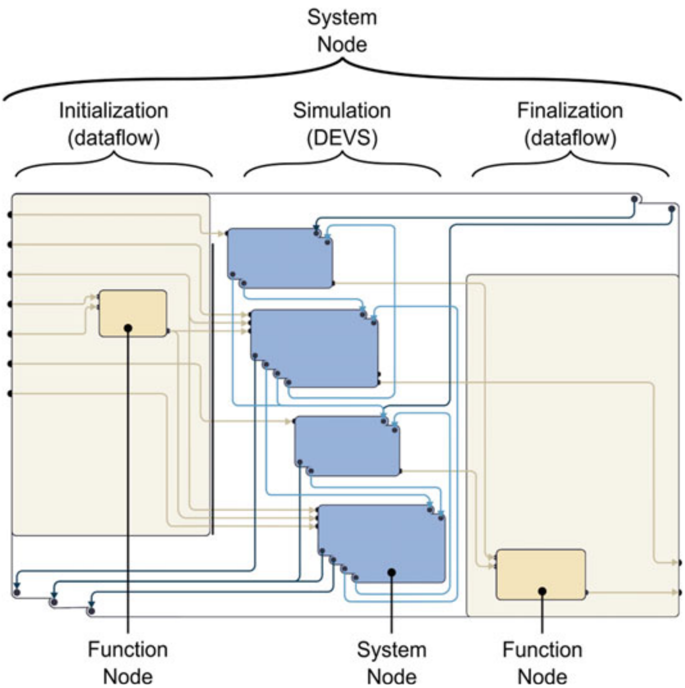


Fig. 8.8 Visual interface mockups combining dataflow and DEVS elements

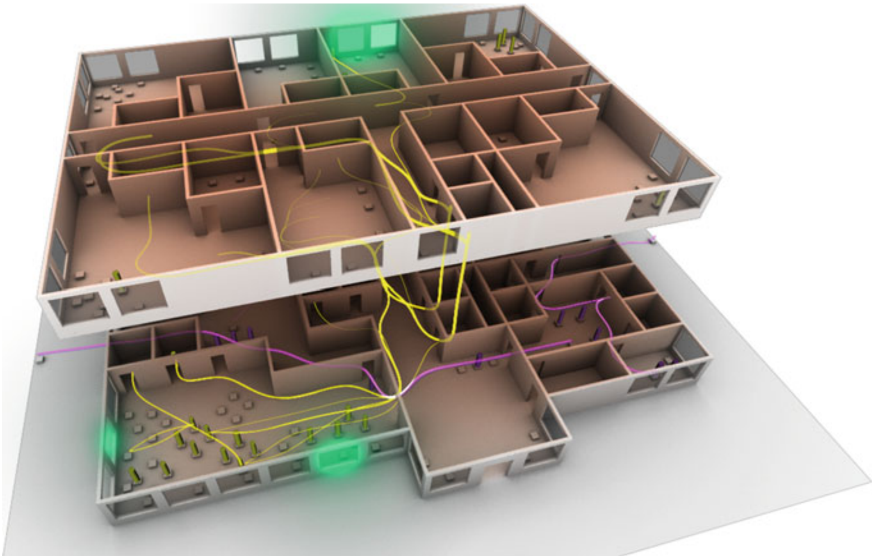


Fig. 8.9 Systems approach proof-of-concept modeled using DesignDEVS and visualized with Autodesk Maya

A key difference between dataflow programming and DEVS is that the latter allows cycles in the node graph. For example, if one of the inner system nodes in Fig. 8.8 represents a building's occupants, and another represents the building's indoor temperature distribution, the two-way relationship between human behavior and building thermodynamics can be established using links from each node to the other. The use of DEVS for this type of scenario was demonstrated by Goldstein et al. (2014) using DesignDEVS. New visualization techniques shown in Fig. 8.9 were researched by Breslav et al. (2014) to visualize the results. The speedlines in the figure animate the movements of a hotel's guests and employees, while glowing effects draw attention to the opening of windows by occupants seeking to improve their comfort level. The state of the windows affects the diffusion of heat through the building, shown as a color gradient on the floor. The indoor temperature then affects the likelihood of additional windows being opened.

The use of DEVS allowed the various simulation algorithms of the Fig. 8.9 model to be rapidly integrated, albeit by modelers with considerable experience with M&S techniques. Visual programming interfaces such as those in Fig. 8.8 may help introduce architectural researchers and building scientists to these scalable practices. Although the approach presented here could be used to model any real-world system, the popularity of dataflow programming among architects enhances its prospects in the realm of building design.

A systems approach offers a new way for researchers to collaborate in pursuit of next-generation simulation-based building design tools. It also represents a strategy for accommodating new design paradigms, in particular the emerging paradigm of

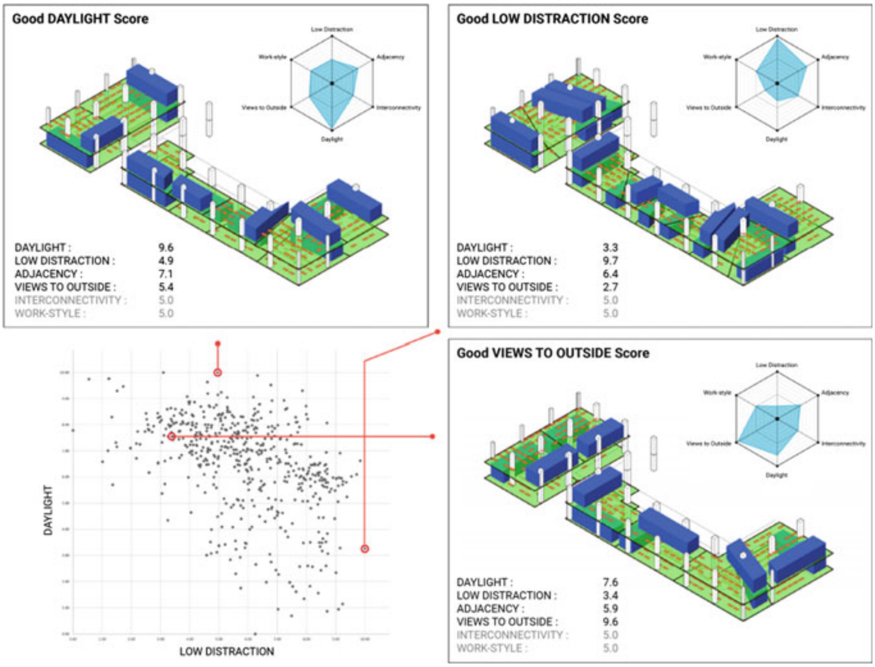


Fig. 8.10 Example of generative design for an actual office environment within the MaRS Discovery District in Toronto, Canada

generative design. Cloud computing and other technological developments have enabled computers to recommend plausible building geometries and systems configurations, primarily by generating and evaluating a myriad possibilities and automatically discarding poor performing options. Along with the closely related topic of multiobjective optimization (Keough and Benjamin 2010), generative design is receiving an increasing amount of attention among architectural researchers. Recently, the technique has been applied to the design of an office layout in order to satisfy several performance criteria including access to daylight, limited potential for distraction, and visual access to the building's surroundings. Figure 8.10 shows three generated layouts selected from a large sample of options.

The generative design project of Fig. 8.10 serves as an informative example for a number of reasons. First, it features geometric analyses that lend themselves well to dataflow programming, and will eventually need to be complemented with simulation. A systems approach combining dataflow with DEVS supports both the implemented analyses and the future simulation algorithms. Second, the results of the analyses are aggregated into a small number of performance metrics, which help inform the next iteration of generated layouts. The dataflow elements at the bottom right of Fig. 8.8 could provide a standard and scalable mechanism for deriving these performance metrics from geometric analyses and simulation results. Third, the project focuses on the experience of occupants in the built environment, a chief concern among architects that is not adequately addressed by current whole building energy modeling tools. Evidently, there is a need to provide architects with software that helps them satisfy objectives related to both human experience and sustainability.

8.5 Conclusion

When one considers the many processes and interactions that take place in and around buildings, as well as the extraordinary impact buildings have on the environment and on how people live their lives, the case for simulation in building design is obvious. Simulation is now heavily used by both architects and engineers in the AEC industry, for a variety of purposes including energy use prediction, structural analysis, and crowd planning. It is also actively researched, with occupant behavior and daylight simulation representing just two of the many current areas of interest. Yet the need for co-simulation developments such as the BCVTB—which is nevertheless an important, pioneering project—speaks to a legacy of large simulation codebases that were groundbreaking in their day but now limit the number of researchers who can effectively collaborate in the development of next-generation building design tools. Complex systems M&S ideas, particularly those that build upon dataflow programming and other techniques familiar to designers, may help architectural researchers and building scientists collaborate toward their common goal of creating more functional, sustainable, and compelling built environments.

Review Questions

1. Buildings account for approximately what percent of electricity use in the United States?
2. What is the difference between a building information model (Fig. 8.1) and an energy analytical model (Fig. 8.2)?
3. Which of these organizations/conferences—ACADIA, ASHRAE, CAAD Futures, IBPSA, Rob|Arch, SimAUD, Smartgeometry—focus primarily on (a) architecture, (b) engineering, (c) both disciplines?
4. What form of visual programming has recently become popular in the architectural design community?
5. The annual cost of heating and cooling a building is an example of a performance metric that could be used as part of a simulation-based architectural design workflow. What other performance metrics could be computed using simulation and applied to improve the design of a building?

Acknowledgements We thank Erin Morrow (Oasys/Arup), Davide Schaumann (Technion), Nathaniel Jones (MIT), Thierry Nouidui (LBNL), David Benjamin (The Living), and Hali Larsen (Autodesk Research) for their assistance in presenting examples of state-of-the-art simulation-based research and tools in building science and computer-aided architectural design.

References

Aasem, E. O., Clarke, J. A., Hensen, J. L. M., Kelly, N. J., MacQueen, J., Negrao C. O R. (1994). Current building systems modelling potential of ESP-r. Proceedings of the International System Simulation in Buildings Conference.

ASHRAE (2004). Standard 90.1, Appendix G.

Autodesk (2017). The Dynamo Primer (for Dynamo v1.2). Online visual programming guide. http://dynamoprimer.com/

Bazjanac, V., Maile, T., Rose, C., O'Donnell, J., Mrazovic, N., Morrissey, E., and Welle, B. (2011). An Assessment of the use of Building Energy Performance Simulation in Early Design. Proceedings of the IBPSA Building Simulation Conference.

Beausoleil-Morrison, I., Macdonald, F., Kummert, M., McDowell, T., Jost, R., and Ferguson, A. (2011). The design of an ESP-r and TRNSYS co-simulator. Proceedings of the IBPSA Building Simulation Conference.

Berkeley, P., Haves, P., Kolderup, E. (2014). Impact of modeler decisions on simulation results. Proceedings of the ASHRAE/IBPSA-USA Building Simulation Conference.

Breslav, S., Goldstein, R., Tessier, A., Khan, A. (2014). Towards visualization of simulated occupants and their interactions with buildings at multiple time scales. Proceedings of the Symposium on Simulation for Architecture & Urban Design (SimAUD).

Brooks, C., Lee, E. A., Liu, X., Neuendorffer, S., Zhao, Y., and Zheng, H. (2007). Ptolemy II – Heterogeneous concurrent modeling and design in Java. Technical Report No. UCB/EECS-2007-7, University of California at Berkeley.

Building Controls Virtual Test Bed. Webpage managed by Lawrence Berkeley National Laboratory (LBNL). Accessed Jan 19, 2017 at https://simulationresearch.lbl.gov/bcvtb.

Building Energy Software Tools Directory (BEST-D). Website managed by IBPSA-USA. Accessed Jan 11, 2017 at http://www.buildingenergysoftwaretools.com/software-listing

Clarke, J. (2001). Energy Simulation in Building Design. Butterworth-Heinemann.

Crawley, D. B., Lawrie, L. K., Winkelmann, F. C., Buhl, W.F., Huang, Y. J., Pedersen, C. O., Strand, R. K., Liesen, R. J., Fisher, D. E., Witte, M. J., Glazer, J. (2001). EnergyPlus: creating a new-generation building energy simulation program. Energy and Buildings, 33(4), 319–331.

Curtis, R. B., Birdwall, B., Buhl, W. F., Erdem, E., Eto, J., Hirsch, J. J., Olsen, K. H., Winkelmann, F., C. (1984). The DOE-2 building energy analysis program. Proceedings of the Asean Energy Conservation in Buildings Conference.

Doore, K., and Vega, D., Fishwick, P. (2015). A media-rich curriculum for modeling and simulation. Proceedings of the Principles of Advanced Discrete Simulation Conference (PADS).

Goldstein, R., Breslav, S., Khan, A. (2013). Using general modeling conventions for the shared development of building performance simulation software.

Goldstein, R., Breslav, S., Khan, A. (2014). Towards voxel-based algorithms for building performance simulation. Proceedings of the IBPSA-Canada eSim Conference.

Goldstein, R., Breslav, S., Khan, A. (2016). DesignDEVS: Reinforcing theoretical principles in a practical and lightweight simulation environment. Proceedings of the Symposium on Theory of Modeling and Simulation (TMS/DEVS).

Hensen, J. L. M., Lamberts, R. (2012). Building Performance Simulation for Design and Operation. Spon Press.

Janak, M. (1997). Coupling building energy and lighting simulation. Proceedings of the IBPSA Building Simulation Conference.

Jones, N. L., Reinhart, C. F. (2014). Irradiance caching for global illumination calculation on graphics hardware. Proceedings of the ASHRAE/IBPSA-USA Building Simulation Conference.

Jones, N. L., Reinhart, C. F. (2015). Validation of GPU lighting simulation in naturally and artificially lit spaces. Proceedings of the IBPSA Building Simulation Conference.

Keough, I., Benjamin, D. (2010). Multi-objective optimization in architectural design. Proceedings of the Symposium on Simulation for Architecture & Urban Design (SimAUD).

Livingston, O. V., Cole, P. C., Elliott, D., B., & Bartlett, R. (2014). Building Energy Codes Program: National Benefits Assessment, 1992–2040. Report prepared by the Pacific Northwest National Laboratory for the United States Department of Energy. 91 pages.

Maleki, M., Woodbury, R., Goldstein, R., Breslav, S. and Khan, A. (2015). Designing DEVS visual interfaces for end-user programmers. Simulation: Transactions of the Society for Modeling and Simulation International, 91(8), 715–734.

Mode Lab (2015). The Grasshopper Primer (3rd edition V3.3). Online visual programming guide. http://grasshopperprimer.com/.

Morrow, E. (2010). MassMotion: Simulating human behavior to inform design for optimal performance. The Arup Journal.

Morrow, E., Mackenzie, I., Nema, G., Park, D. (2014). Evaluating three dimensional vision fields in pedestrian microsimulations. Proceedings of the Pedestrian and Evacuation Dynamics Conference.

Nouidui, T. S., Wetter, M., Zuo, W. (2013). Functional Mock-Up Unit import in EnergyPlus for co-simulation. Proceedings of the IBPSA Building Simulation Conference.

Schaumann, D., Kalay, Y. E., Hong, S. W., and Simeone, D. (2015). Simulating human behavior in not-yet built environments by means of event-based narratives. Proceedings of the Symposium on Simulation for Architecture & Urban Design (SimAUD).

Ward, G.J. (1994). The RADIANCE lighting simulation and rendering system. Proceedings of the SIGGRAPH Conference, 459–472.

Wetter, M. (2011). Co-Simulation of building energy and control systems with the Building Controls Virtual Test Bed. Journal of Building Performance Simulation (JBPS), 4(3): 185–203.

Woodbury, R. F. (2010). Elements of Parametric Design. Routledge.

Zeigler, B. P., Praehofer, H., Kim, T. G. (2000). Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems (2nd edition). Academic Press.

Author Biographies

Rhys Goldstein is a simulation expert in the Complex Systems group at Autodesk Research. His work straddles the disciplines of simulation theory and architectural design, and he has contributed as a researcher and conference organizer in both areas. Rhys and his colleagues received two overall Best Paper Awards: one at the 2010 IBPSA-USA Conference, for the customization of simulated human behavior in buildings; and one at SpringSim 2016, for a simulation development environment called DesignDEVS. Rhys served as the 2014 Co-Chair and 2015 Program Chair of the Simulation for Architecture and Urban Design (SimAUD) Symposium, and as Program Co-Chair of the 2017 TMS/DEVS Symposium. He is also an Associate Editor of the SIMULATION journal.

Azam Khan is Director, Complex Systems Research at Autodesk. He is the Founder of the Parametric Human Project Consortium, SimAUD: the Symposium on Simulation for Architecture and Urban Design, and the CHI Sustainability Community. He is also a Founding Member of the International Society for Human Simulation and has been the Velux Guest Professor at The Royal Danish Academy of Fine Arts, School of Architecture, at the Center for IT and Architecture (CITA) in Copenhagen, Denmark. Azam has published over 50 articles in simulation, human–computer interaction, architectural design, sensor networks, and sustainability. His Toronto team is currently developing a new experimental simulator to explore big simulation as a component of eScience, and his New York team, The Living, performs award winning design research in advanced architectural projects.

Part III
Natural Sciences

Chapter 9
Simulation-Based Science

Toward Cognitive Generative Architectures for Simulation-Driven Discovery

Levent Yilmaz

Abstract The use of computational simulation in science is now pervasive. However, while model development environments have advanced to a degree that allows scientists to build sophisticated models, there are still impediments that limit their utility within the broader context of the scientific method. Despite availability of effective tools that assist scientists in routine aspects of scientific workflow management and analytics, other steps, including explanation, evidential reasoning, and decision-making, continue to limit the process of causal reasoning in knowledge discovery and evaluation. This chapter examines the types, functions, and purposes of models in relation to the scientific method, identifies the issues and challenges pertaining to information abstraction and cognitive support for computational discovery, and delineates a model-driven cognitive systems approach for simulation-based science.

Keywords Model-driven science · Cognitive computing · Model-driven engineering · Intelligent agents · Learning · Autonomous models · Self-awareness · Cognitive models · Mediation · Experiment management · Explanatory coherence · Domain-specific languages · Computational discovery · Cognitive coherence · Parallel constraint satisfaction · Model-based reasoning · Simulation-based science · Dynamic data-driven application · Generative modeling · Cognitive systems

9.1 Introduction

The process of causal reasoning in scientific knowledge discovery and evaluation involves various steps, including identification of gaps in the current state of knowledge, generating inquiries for investigation based on current priorities and the

---

L. Yilmaz (✉)
Computer Science and Software Engineering, Auburn University, 3116 Shelby Center, 36849 Auburn, AL, USA
e-mail: yilmaz@auburn.edu

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_9

state of knowledge, designing, and executing plans for new data and evidence gathering, analyzing and interpreting results, and revising hypotheses (Bunge 1998; Darden 2002; Gelfert 2016). To support these steps transparent cognitive models are needed to assist computational discovery while serving as effective communication tools that explain observations and make recommendations for model building (Honavar et al. 2016). Moreover, cognitive models should also be psychologically plausible, so that they not only account for, but when necessary, steer the cognitive activities of the scientist.

To examine the role of models, in general, and cognitive computing, in particular, we examine the nature and characteristics of scientific behavior. Following the characterization of the processes of scientific discovery, we outline the types, functions, and purposes of models in relation to computational discovery. To advance the state of the art, we put forward a framework for Model-Driven Discovery (MDD), which is based on the premises of the Dynamic Data-Driven Application Systems (DDDAS) paradigm (Darema 2004). By viewing MDD from the perspective of DDDAS, model discovery and experimentation are considered as adaptive autonomous processes that evolve as learning takes place. A dual search process over the model and experiment technical spaces aims to improve the accuracy of models in explaining and understanding of scientific phenomena. In the context of the DDDAS perspective, models can be viewed as mediators (Morrison and Morgan 1999) between the theory and data, while orchestrating multiple views and different interpretations of the target system.

In the rest of the chapter, we first overview the elements of scientific behavior, including the types of scientific knowledge and activities. As part of the background section, we examine the state of the art in the use of computational models and simulations, as well as the types, functions, and purposes of models in scientific discovery. In Sect. 9.3, we highlight emerging issues and opportunities in accelerating scientific discovery. To address the highlighted issues, we propose, in Sect. 9.4, a conceptual framework that views models as adaptive learning agents that leverage cognitive computing techniques to guide scientists in hypothesis generation and experiment design. Building on the theory of Model-Driven Engineering (MDE), Sect. 9.5 delineates the use of advanced principles of MDE for constructing scientific models. Section 9.6 introduces a cognitive modeling strategy that builds on the proposed conceptual foundations. Section 9.7 concludes with a summary of the contributions of the article.

9.2 Background

The main goals of science include providing explanations, making predictions, developing instrumental applications of scientific findings, and exploration to improve our ability to understand the world around us (Gelfert 2016). Model-based explanation, prediction, and understanding have been influential as a strategy for providing guidance to scientific activities and the production of scientific knowledge.

9.2.1 *Scientific Knowledge and Activities*

The creation of scientific taxonomies, laws, and theories, as well as their revision based on new information is at the core of the scientific method (Klahr and Simon 1999). These scientific knowledge structures are generated and manipulated by scientific activities.

9.2.1.1 Scientific Knowledge Structures

There are various forms of knowledge involved in scientific process. These include taxonomies, hypotheses, scientific laws, experiments, and theories (Bunge 1998).

- **Taxonomies** classify and organize knowledge in terms of concepts, attributes, and relationships to facilitate systematic understanding of a problem domain (Bunge 1998). Taxonomies include specialization and generalization relationships among concepts to account for commonalities. As the scientific discipline advances, the ontology of the domain becomes increasingly specialized and reflect the level of accuracy and specificity gained by the scientific method. On the other hand, as experience is gained, discovered entities with common attributes are generalized to produce abstract and high-level knowledge that can facilitate drawing broader inferences and general applicability.
- **Hypotheses** refer to ideas related to experience, expectation, or observations in the form of assumptions, which are used as building blocks in the construction of scientific explanations and testable, falsifiable models. As evidence builds, hypotheses are upgraded into laws and theories (Langley 2000).
- **Laws** specify precise numerical or qualitative relations and patterns among observed variables, events, and objects.
- **Theories** use the terminology of the problem domain, which is defined by its conceptual taxonomy, to make statements about the structure and processes of a phenomena by connecting and relating laws/patterns and hypotheses into a unified theoretical account.
- **Experiments** involve experimental design and manipulation of the knowledge structures and the context to test hypotheses by examining their consequences under specified conditions.

Figure 9.1 presents a conceptual model of scientific knowledge structures in relation to models. Theories are based on hypotheses that are upgraded into laws when they become effective in explaining the evidence observed in the target phenomena. Models are abstractions that view a system from the perspective of a theory, aiming to account for expected or observed regularities. Therefore, the model mediates between the data and the theory to determine a coherent representation of hypotheses that fit together to account for both the theory and the data. Of the various elements that make up the model, some features are derived from the

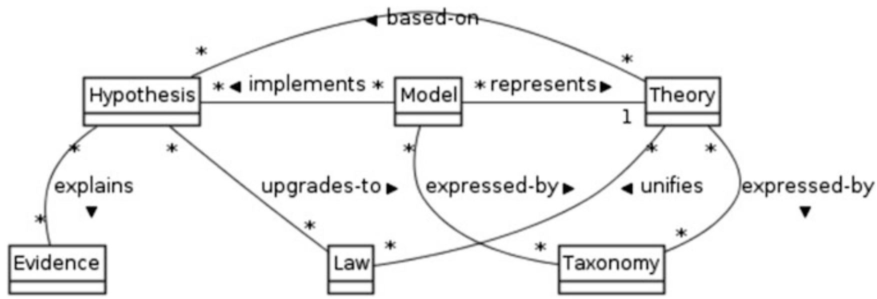


Fig. 9.1 Scientific knowledge structures

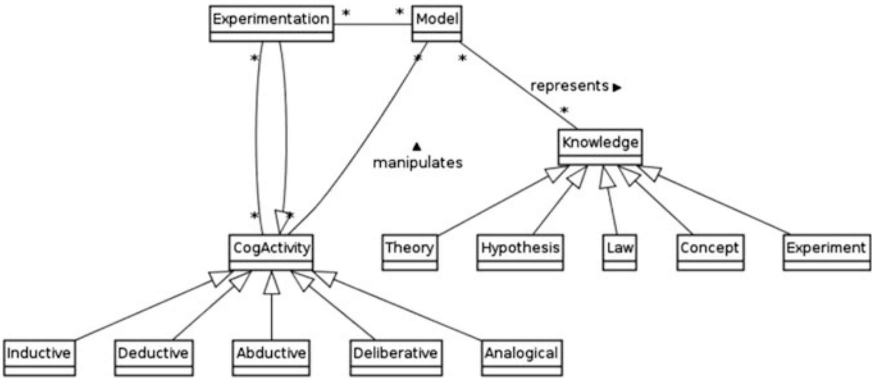


Fig. 9.2 Cognitive activities

theory, while others originate from data and other pragmatic considerations specific to the purpose of the study.

The mediation process suggests that model construction is a complex integration activity. However, the mediation process brings together not only theoretical constructs and expected behavior, but also analogies, metaphors, modeling concepts and techniques, theoretical notions, etc. Therefore, the process of modeling involves (1) molding the ingredients of the model, including hypotheses, toward a unified and coherent knowledge representation and (2) calibration of the parameter and experiment spaces for the purpose of integrating all ingredients. As such, the assessment of models in scientific knowledge generation can be construed from the perspective of how well they can be flexibly integrated and adapted to bring coherence to constituent and align with the empirical context.

9.2.1.2 Scientific Activities

Knowledge generation activities involve distinct categories of cognitive processes shown in Fig. 9.2 and can be classified under inductive, deductive, and abductive, deliberative, and analogical activities (Bunge 1998; Langley 2000). Inductive activities include the formation and learning of concepts and taxonomies, inductive law discovery, and inductive theory generation. Learning of taxonomies is an automated abstraction, clustering, and concept generation process that can take advantage of machine learning and data analytics techniques. Inductive law generation or revision involves generalization of data into meta-models that cover the observations in terms of precise invariants, functions, or relations.

Inductive theory generation integrates the laws and connects them into a unified account. Data-driven scientific paradigm (Hey et al. 2009) often focuses on inductive data-driven techniques for law and theory generation; whereas simulation enables theory-driven deductive and abductive processes.

The deductive aspect stems from models being interpreted by simulation engines to deduce through computational processes the consequences of the assumptions embodied in the models. Scientists often derive predictions from models, or deduce laws in the form of patterns and regularities from the theoretical principles specified in models. Deliberative activities include prescriptive decision-making such as selection of theoretical principles as well as experiment design strategies to reduce the uncertainty among competing hypotheses. Analogy is also a fundamental activity, for it involves comparing and relating representations on the basis of structure and behavior for the purpose of understanding and explanation.

As shown in Fig. 9.3, cognitive activities support various scientific activities such as explanation, prediction, and exploratory understanding. *Prediction* takes a meta-model or a model to produce an estimate. Then the *explanation* process connects a theory to a law in terms of a model and its underlying mechanistic hypotheses, which generate the invariant defined by the law. The model represents the explanatory framework by which the theory is connected to the laws and the evidence observed. If explanation fails, the anomaly triggers revision of the theory or the law. Explanation also relies on abductive cognitive activities that allow the scientist to posit new unobserved hypothetical assumptions rather than deduction of emergent behavior solely via simulation of given premises.

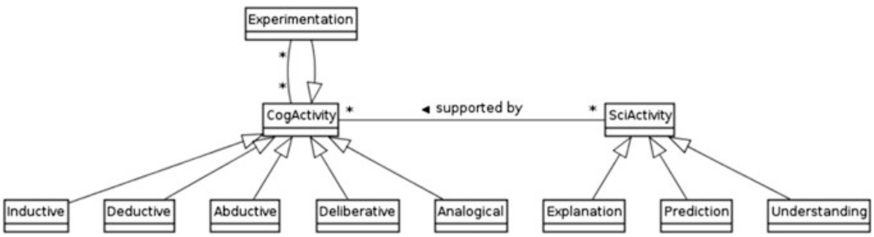


Fig. 9.3 Scientific activities

9.2.2 *Types, Functions, and Exploratory Purposes of Models*

The Stanford Encyclopedia of Philosophy provides an extensive list of model-types in the context of philosophy of science: computational models, scale models, heuristic models, meta-models, theoretical models, scale models, probing models, phenomenological models, didactic models, formal models, iconic models, analog models, and instrumental models. From the perspective of simulation-driven science, we focus on the functional characterization of computational models in terms of how they carry scientific knowledge. We categorize models with respect to degrees of representational detail and support for experimentation. Based on the degree of representation, models can range from purely data-driven models to theory-driven models. On the other hand, depending on the support that models lend to experimentation, they can be categorized as informational or pragmatic (Gelfert 2016).

Figure 9.4 classifies models into four broad categories using the degree of representation and experimentation dimensions. Data-driven models (Hey et al. 2009) can be classified as *phenomenological* and *instantial* models. Phenomenological models are mathematical expressions that attempt to explain the relations among observed variables under controlled experiments (Kleijnen 2008). A phenomenological model can be consistent with a fundamental theory, but it is not derived from theory. For instance, regression models are phenomenological models. Instantial models represent functions or relations that may be associated with general purpose domain-independent theoretical constructs, but they are derived neither from the theory about the phenomena nor experiments. Instantial models can be calibrated with data to make predictions or postdictions about observations, but they are theory-free constructs.

Theory-driven models which are intended to explain or understand a phenomenon within the context of an experimental framework are known as

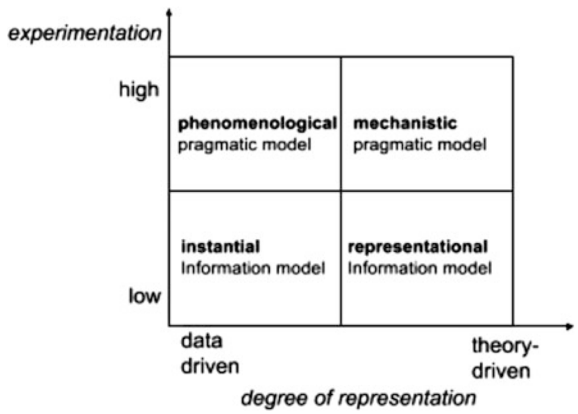


Fig. 9.4 Functional categorization of models

mechanistic models (Darden 2002) or causal microscopic models. Such models are comprised of proper representational abstractions that facilitate effective and efficient exploration to answer specific research questions in a pragmatic manner, and there are well-defined mechanism discovery protocols associated with such theory-driven modeling perspective. Representation-based informational models are connected to the target phenomena in terms of constructs that are directly related to target's entities and processes. In informational models, the representational aspects are made more central compared to the pragmatic features that relate to the purpose of the model.

Models can be used for different purposes, including explanation, prediction, and exploratory understanding (Gelfert 2016). Exploratory models can support both specific and divergent sense of exploration (Davis 2000, 2015). If exploration focuses its attention on a salient research problem, the process converges upon a specific question or detail. This view contrasts with the divergent sense of exploration, which is not directed toward specific object or question. Instead, the inquiry seeks novelty or surprising findings on its own sake. Divergent exploration can lead to a narrower view and switch to a mode of specific exploration as more insight is gained. Exploratory mode of modeling includes the following activities:

- examine the behavior of a model across a wide range of experimental conditions,
- determine which parameters are prominent and have impact on the targeted attributes,
- discern invariants and state them in the form of hypotheses,
- explore representations by which these rules can be formally defined.

As exploratory tools, models can serve different purposes, including the provision of a baseline starting point for future inquiry, proof-of-concept demonstration, and potential explanations (e.g., hypotheses) for a specific phenomenon. Exploratory modeling is neither limited to application of fundamental theory nor constrained by observational data, but typically involves a constructive modeling effort. These models are not intended to generate testable empirically adequate predictions. Instead, the objective is to generate insight into the phenomena. For instance, Lotka–Volterra equation models facilitate understanding the dynamics of discrete populations, whose size is measured as integers using continuous differential equations. The model demonstrates, as a proof-of-concept, that periodic oscillations in the size of prey–predator populations may emerge purely internally. The model brings new avenues for mathematically modeling the dynamics of populations.

An important value of scientific models comes from their explanatory roles, especially when models are defined in terms of mechanisms that represent *why* and *how* the observed behavior emerges. Explaining a phenomenon often involves construction of a mechanistic model (Darden 2002) that is subsumed by a theory and its fundamental axioms and hypotheses. On the other hand, in the absence of a theory, exploratory models help scientists devise potential explanations. The model helps envisage mechanisms that, if true, would give rise to behavior that constitute the explanandum.

With their support to explanation, prediction, and exploratory understanding activities of scientific knowledge generation, models support science as *mediators*, *contributors*, and *epistemic tools* (Gelfert 2016). In the case of mediation, a model aims to fit together theoretical and evidential constructs so as to converge to an explanation of observations, or make predictions under various circumstances. The molding of the ingredients of the model to fit theory and data suggests the autonomous characteristic of a model. In the absence of a fully defined theory, scientists construct models in an attempt to learn more about hypothetical features, the realization of which allow studying their theoretical consequences. Alternatively, a model can be constructed in a specific manner, which facilitates application of idealized and abstract laws. Models can also serve as the objects of inquiry in an exploratory mode, especially when there is lack of access to potential target systems or phenomena.

Models are also contributors of knowledge, for they do not merely integrate or mediate existing theoretical and empirical elements. New elements and constructs are formulated to not only conceptualize theoretical mechanisms, but also integrate them in ways that are consistent with the phenomena. Through the construction and manipulation of hypothetical features, models contribute as *epistemic tools* that facilitate generation via exploration of and justification via experimentation with knowledge. However, to facilitate acquisition of scientific knowledge, models should not only be connected with a valid and objective relation to the phenomena of interest, but also afford cognitive access to the information it contains. This dual relation of models to the scientists and the target phenomena suggest two distinct aspects, which relate to *transparency* and *veridicality*. The transparency aspect of a model refers to the phenomenology of the interaction afforded by the model's representation, whereas veridicality focuses on the degree to which the model's information space is connected to the target system.

9.3 Issues and Challenges

The use of computing as a formal framework for accelerating various aspects of scientific inquiry has become a considerable interest (Honavar et al. 2016). Specifically, the role of computational thinking in the acquisition, organization, verification, analysis, reasoning, integration, and communication of scientific artifacts such as data, models, theories, hypotheses, and experiments is well recognized. As an exploratory instrument, simulation models have been particularly effective in the context of mechanism-centered perspective of scientific discovery. As computation increasingly becomes the language of science, we recognize emerging issues and challenges to keep pace with the information-centric aspects of knowledge discovery.

9.3.1 Information Processing Abstractions

The use of algorithmic information processes in developing models of scientific phenomena is becoming increasingly common. Computer simulations have become instruments of epistemic inquiry in a wide range of domains spanning from natural sciences to social and artificial sciences. Further developments in the use computational modeling of information processes underlying scientific phenomena calls for advancements, including, but not limited to the following challenges:

- Conceptualization of information abstractions in a given scientific domain requires methods and tools that streamline the derivation and formalization of such abstractions. Algorithmic and conceptual abstractions of natural entities, relations, and processes of interest in specific scientific domains need to be derived from their extant literature.
- Scientists may not have expertise in general purpose programming environments. Therefore, high-level domain-specific modeling languages are necessary to make models in specific domains expressible in terms of the terminology of the domain (Teran-Somohano et al. 2015; Yilmaz et al. 2016).
- Simulation models and the outputs of computational discovery systems that leverage such models need to be communicated to domain scientists. Information processing abstractions such as state and activity models, Bayesian nets, and probabilistic decision models used in the specification of models may differ from the formalisms used by domain scientists. Model management and automated transformation technologies can play a role in bridging the gap between the engineering space of simulations and the technical space of the scientific domain.
- Discovery systems need to account for domain knowledge and constraints associated in a domain. Formal methods and analysis techniques need to be leveraged to assure that models are not only syntactically well-formed, but also adhere to semantic constraints that limit the mechanistic hypotheses that can be posited by the scientist while using the modeling system.
- Discovery systems need to produce or use models that go beyond description to provide explanations of evidence in terms of mechanisms. Yet, the use of data analytics often aims to reveal descriptive regularities. However, scientific activities are primarily concerned with model-driven explorations, explanations, and predictions that account for or embody accounts of mechanisms that incorporate theoretical variables, objects, and processes. Deeper mechanistic understanding of scientific phenomena requires generation, revision, and testing of such mechanisms. Model development environments that facilitate variation and revision of mechanistic hypotheses are needed to adapt models as learning takes place.
- Experimentation is a critical aspect of the process of scientific inquiry. Yet, simulation environments often lack support for designing and managing experiment models along with the models of the target system. Syntactic and semantic support is needed for explicit specification of experiment models

(Teran-Somohano et al. 2015) as well as the execution and management of models.

- Scientific phenomena often include multiple aspects and levels that require provision of multiple distinct formalisms (Yilmaz et al. 2007) appropriate for each aspect. The use of hybrid models (Mosterman 1999) and formalisms along with mechanisms that facilitate seamless interoperation and couplings across formalisms will increasingly become important.

The challenges outlined above are concerned with advances in the theory and methodology of modeling and simulation to better respond to characteristics and nature of scientific activities and knowledge structures. The use of such advanced modeling and simulation environments is necessary, but not sufficient to support specific cognitive activities associated with scientific reasoning. In the next section, we expound on the role of cognitive computing and delineate ways to incorporate them into simulation-driven scientific discovery systems.

9.3.2 *Cognitive Issues in Scientific Discovery*

Cognitive computing is concerned with the development of computational models and techniques that study the human mind and simulate human thought processes, including learning, decision-making, planning, problem solving, reasoning, and explanation. The uses of cognitive models can augment and amplify scientists' capabilities to formulate questions, generate hypotheses, formulate questions, plan and devise experiments, and analyze results to facilitate learning from experience to revise both the hypotheses and experiments so as to improve the discovery process.

Figure 9.5 highlights four major modes of cognitive assistance that can contribute to a model-driven discovery cycle. Assistance in the form of automation of the design and execution of experiments as well as model generation and revision

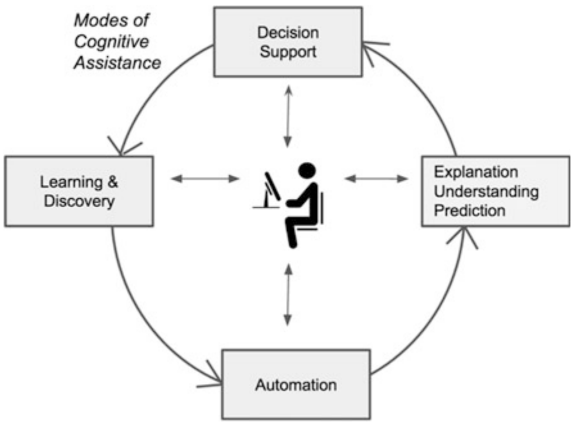


Fig. 9.5 Modes of cognitive assistance

enables scientists to focus more on the goals and objectives of the study rather than routine aspects of platform management. Among the issues and challenges that can be addressed by the use of cognitive computing methods and techniques include the following.

- Problem formulation requires development of domain models based on the common core concepts and terminology of a problem domain. Publications disseminated in the extant literature can be analyzed using natural language understanding methods to automate derivation of conceptual models and taxonomies, which provide the basis for information processing abstractions.
- The need for mechanistic explanation of scientific phenomena requires constructing models that embody causal mechanisms that produce, sustain, or prevent targeted behaviors. Automated support in the exploratory search for mechanistic hypotheses can leverage heuristic search techniques, as well as abductive, probabilistic (Pearl 2014), and analogical reasoning methods to postulate model revisions.
- Testing the consequences of hypothesized mechanistic models requires deciding which variables to measure, as well as designing and prioritizing experiments to improve information gain by reducing uncertainty across hypotheses while adhering to principles of reliable and valid experimentation. Cognitive assistance can also help determine the marginal utility of experiments and provide support for comparing alternative experiments and devising scientific workflows.
- Execution and orchestration of simulation experiments, possibly on distributed platforms, and collecting data to evaluate hypotheses requires transparent access and instrumentation into software models to trace variables that are of interest to selected experiments. Experiments should be coupled to model representation to support the observation process while also improving controllability of the simulation.
- The data collected through simulation experiments need to be abstracted into evidence, patterns, and observed regularities for the purpose of evaluating hypotheses. Inductive learning and generalization methods, followed by deductive formal methods (e.g., model checking with temporal logic) can be used to determine whether or not the hypothesized mechanistic assumptions exhibit the desired behavior.
- The discovery system needs to provide support for learning from experimentation so as to discriminate across competing hypotheses. For instance, statistical machine learning techniques can lend support to generating causal probabilistic networks among hypotheses and evidences to provide a quantitative explanatory framework for their assessment. Cognitive theories and models of explanatory coherence can be leveraged to discern acceptable hypotheses while rejecting those that cannot serve as plausible explanations.
- The output of existing learning techniques is difficult to communicate to disciplinary scientists. Advances in cognitive systems are necessary to produce explainable models and present results via an explanation interface, which is

guided by cognitive and psychological theories of effective explanations. Alternative machine learning models can be developed to learn structured, interpretable, causal models. Moreover, model induction techniques can be used to infer meta-models that provide approximate explainable models of causal dependencies among the scientific knowledge structures.

- Following the analysis of the results, if necessary, mechanistic hypotheses need to be revised, and these revisions, including new behavioral mechanisms need to be transferred into the model space to start a new cycle of experimentation. This requires revision of experiment models to bring focus and generate new plans. Alternative experiment plans, guided by the current goals, can be explored until when new goals emerge as a result of the revision of hypotheses and the emergence of new evidence.

9.4 Models as Autonomous and Adaptive Cognitive Agents

The use of models as exploratory instruments that evolve into a plausible explanatory or predictive model suggests that it is reasonable to view models as Dynamic Data-Driven Application Systems (DDDAS) (Darema 2004). In this view, as a mediator between theory and data, model development includes seeking coherence among mechanistic hypotheses that govern a model's behavior.

9.4.1 Modeling as a Dynamic Data and Theory-Driven Process

Figure 9.6 illustrates the mediation role of models between theory and data. Theoretical principles are leveraged to construct mechanistic hypotheses in the form

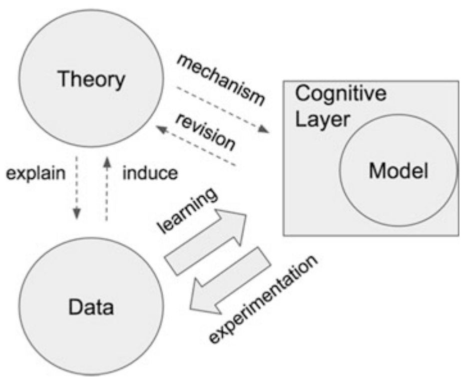


Fig. 9.6 Models as mediators between theory and data

of behavioral rules that embody the specification of model elements. The construction of the model is followed by the experimentation process through simulation. Targeted instrumentation of the model results in observed simulation data that forms the basis for learning about the efficacy of the hypothesized mechanistic and phenomenological assumptions.

The dynamic mutually recursive feedback control loop between the model and data refers to adaptive learning of a model and its role in steering both the instrumentation and the experimentation process. The data gathered through experimentation is analyzed to make decisions about model representation. Model revisions can in the long term result in theory revision. As a consequence, the theoretical principles that may originally be induced from limited field data become increasingly accurate in its explanatory power based on growing set of new data, including those generated by simulation.

The strategy presented above is akin to the DDDAS paradigm (Darema 2004), which promotes incorporation of online real-time data into simulation applications to improve the accuracy of analysis and predictions. As a methodology, DDDAS aims to enhance application models by selectively imparting new observed data or deduced knowledge into the model so as to make it congruent with the evolving context. Moreover, refined models can then be used to control and guide the measurement process. The ability to guide the measurement and instrumentation processes is critical when the measurement space is large. By focusing the measurement process over a focused subset of data, the methodology reduces both the cost and the time to collect data, while also improving quality and relevance.

Enabling the DDDAS perspective in model and experiment management requires advancements in variability management in model representation under uncertainty, system interfaces to instrument the simulation for data gathering and analysis, and incorporates both the data and results of data-driven inferences and decisions back to the model's technical space. The provision of run-time models and run-time dynamic model updating (Yilmaz and Ören 2004; Blair et al. 2009) are critical features for closing the loop.

9.4.2 A Generic Architecture for Models as Cognitive Agents

As an active entity, models with cognitive capabilities provide features that overlay the simulation model and augment it to support computational discovery. In this view, a model is construed as a family of models that evolve as learning takes place while being sensitive to the goals of a particular study. As such, models need to be designed with variability management (Bosch et al. 2015) in mind to support seamless customization and to address a variety of experiment objectives, especially when the target system has multiple facets.

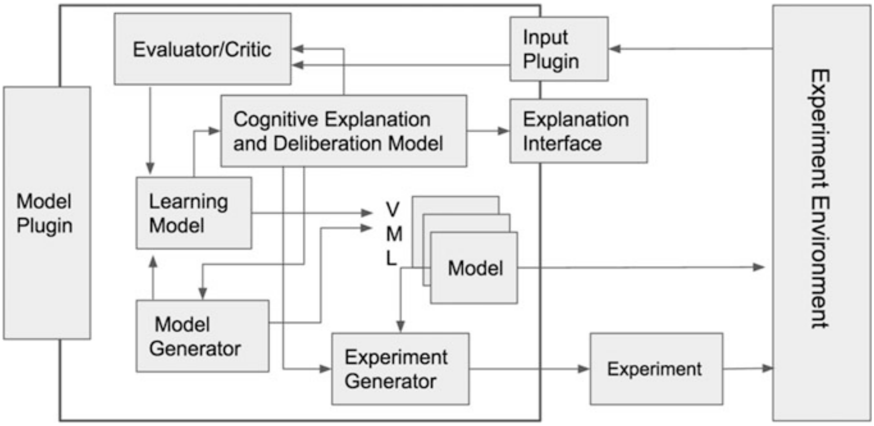


Fig. 9.7 Generic reference architecture

Figure 9.7 illustrates the building blocks of an active model that is coupled to an experimentation environment to maintain a mutually beneficial and adaptive feedback between theory and data. Theoretical constructs are characterized by the *Variability Management Layer* (VML) and the models that encapsulate the mechanistic hypotheses, principles, and constructs underlying the theory. A family of models is defined in terms of features (Kang et al. 2002; Oliveira et al. 2013) that can be configured to synthesize alternative models. Features define mandatory or optional standard building blocks of a family of models along with their interrelationships. Variability management via feature models is a common strategy in the product-line engineering practice to automate the construction and derivation of products. A variability management language should enable the specification of a variability model and its relation to model representation in the host simulation modeling language. The selection of variants in the form of features triggers specific actions that customize a model by adding, removing, or updating functional units. Such an approach views a model in terms of domain-specific objects and configuration scripts that derive model instances that adhere to domain constraints.

An input plug-in transforms raw experiment data into a format that can be analyzed using the *Evaluator/Critic* component. The evaluator component can be as simple as a filter that further abstracts the data. However, to provide cognitive assistance, a sophisticated model can provide support in terms of formal methods such as Probabilistic Model Checking (Kwiatkowska et al. 2002) to determine the extent to which expected behavior is supported by the simulation data. Alternatively, the evaluator can perform hypothesis testing to discern the degree of support that model mechanisms provide to hypotheses under consideration. Those mechanisms that lend significant support to the evidence are retained, while others are revised or declined from further consideration.

The learning model uses the results of the evaluator to update the confidence levels of competing hypotheses. For example, a Bayesian Net model can revise conditional posterior probability estimates using the Bayes' rule in terms of priory probabilities and the observed evidence. Alternatively, cognitive models such as explanatory coherence (Thagard 1989), which is briefly discussed in Sect. 6, can be used to acquire, modify, reinforce, or synthesize hypotheses to steer the model-driven discovery process. This incremental and iterative strategy is coordinated by a mediation process that governs the interaction between the technical spaces of models and experiments.

9.4.3 The Mediation Process

Based on Klahr's (2002) framework on scientific discovery, the mediation process involves three main components that control the entire process from the initial formulation of hypotheses, through their experimental evaluation, to the decision that there is sufficient evidence to accept a hypothesis. As shown in Fig. 9.8, the three components are Search Hypothesis Space, Test Hypothesis (Search Experiment Space), and Evaluate Evidence. The output from the Search Hypothesis Space is a fully nspecified hypothesis, which provides input to the Test Hypothesis phase that involves simulated experiments, resulting evidence for or against the hypothesis. Evaluate Evidence decides whether cumulative evidence warrants acceptance, rejection, or further consideration of the current hypothesis.

Search Hypothesis (Model) Space: This process has two components. The first component generates the structure and broad scope of the hypothesis. The second component refines the structure, making it instantiable.

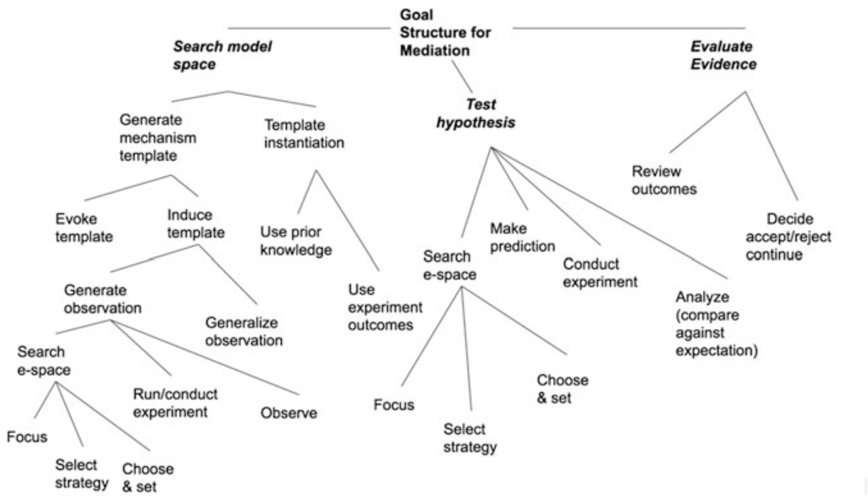


Fig. 9.8 Searching the mechanism/hypothesis space

Generating a mechanism template can be addressed by either invoking or inducing a template. Invoking a template is based on searching existing specifications so as to construct a frame. Prior knowledge or Domain-Specific Language (DSL)-based specification can play a role here. In cognitive science, several mechanisms have been proposed to account for the way by which initial mechanism templates can be identified. One can start with the initial set of mechanistic hypotheses to construct the initial search space. Alternative approaches can be used to iteratively and incrementally build the hypothesis space. Among these techniques include analogical mapping, heuristic search, priming, and conceptual combinations. Abductive reasoning is also a common strategy for creating new plausible hypotheses. When it is not possible to evoke a mechanism, a new mechanism frame can be induced from a series of outcomes. The first process involves generating an outcome through experimentation. The second process uses the data as input to generalize over the outcomes to induce a frame.

The second major component of the Search Hypothesis Space goal is the *Schema* (*Template*) *Instantiation* process. Its purpose is to take a partially instantiated mechanism and assign specific values to generate a fully specified hypothesis. Instantiating a mechanism template involves either using prior knowledge or using specific experimental outcome. If there exist outcomes extracted from previous experiments, they can be adapted and reused in the new context. Alternatively, the Generate Outcome goal can be used to produce empirical results solely for the purpose of determining mechanism's parameter values, facilitating the refinement of a partially defined hypothesis. Early in the course of experimentation, prior knowledge is used to assign values, whereas using experimental outcomes is more likely to be used in the later phases of the iterative discovery process. If all the identified values are tried and declined, then the mechanism template needs to be dropped, and the process returns to the Generate Mechanism goal.

The Generate Observation/Outcome goal appears multiple times in the goal hierarchy of the Search Hypothesis Space. The first appearance is when simulated outcomes are generated in order to induce a mechanism template, and the second occurrence is when the hypothesized mechanisms are instantiated with specific values that facilitate derivation of the complete specification for the mechanisms. Each time the *Generate Outcome* goal is activated so is the Search Experiment Space goal. The experiments are designed by the *Search E-Space* goal. The experiment space search should be able to focus on those aspects of the situation that the experiment is intended to elucidate and explain. Discriminating among competing hypotheses is one of the critical functions of the *Focus* sub-goal. Once a focal aspect has been identified, the *Select Strategy* sub-goal chooses specific independent and control factors according to the search priority and preferences implied by the *Focus* goal. This is similar to the Means-Ends Reasoning and heuristic search/optimization strategies that are often used to sweep a state space to achieve a particular objective. The Choose & Set sub-goal assigns specific values to independent and control variables so as to facilitate the application of the Conduct Experiment and Observe sub-goals.

Search Experiment Space (**Test Hypothesis**): This goal aims to generate an experiment that is appropriate for the current set of hypotheses being examined, to make a prediction by running the simulation experiment, and matching the outcome to expected behavior/evidence. The *Search Experiment Space* component produces an experiment. Conducting the experiment involves execution of the simulation experiment and may require distributing the replications across multiple machines to improve the performance. The *Analyze* (*Compare*) goal aims to provide a description of the discrepancy between the evidence/expected behavior and the actual outcome. The comparison can involve statistical methods such as ANOVA analysis as well as formal methods that leverage Model Checking. The outcomes from simulation replications can generate outcomes that can be generalized as a (Hidden) Markov Model to that can formally verified against the finite state verification patterns. When completed, the *Test Hypothesis* goal generates a representation of evidence for or against the current hypotheses. This outcome is then used by the *Evaluate Evidence* component.

Evaluate Evidence: This component aims to determine whether or not the cumulative evidence gleaned from the results of experiments warrant the acceptance or rejection of competing hypotheses. Various criteria can be used to evaluate evidence and hypotheses. These include plausibility, functionality, parsimony, etc. In the absence of hypotheses, experiments can be generated by moving around the experiment space. During the Evaluate Evidence phase, three general outcomes are possible. The current set of coherent hypotheses can be accepted, it can be rejected, or it can be considered further. In the first case, the discovery process stops. If the hypothesis is rejected, then the system returns to the Search Hypothesis Space, which can trigger two possible activities. If the entire mechanism template (frame) is rejected, then the system must attempt to generate a new mechanism. If the *Evoke Mechanism* goal cannot be satisfied or is unable to find an alternative mechanism, then the system will recourse to the *Induce Mechanism Template* sub-goal, which requires running simulation experiments to generate outcomes that can be generalized via induction to synthesize mechanistic hypotheses. Having induced a new mechanism frame, or having returned from Evaluate Evidence with a frame needing revised instances with new values, the system resumes with *Template Instantiation*. If prior knowledge is not applicable or available, here, too, the system may require running experiments to generate outcomes and to make value assignments.

9.5 A Computational Strategy to Support Simulation-Driven Discovery

Model-Driven Engineering (MDE) (Völter et al. 2013) has emerged as a practical and unified methodology to manage complex simulation systems development by bringing model-centric thinking to the fore. The use of platform independent domain models along with explicit transformation models facilitates deployment of

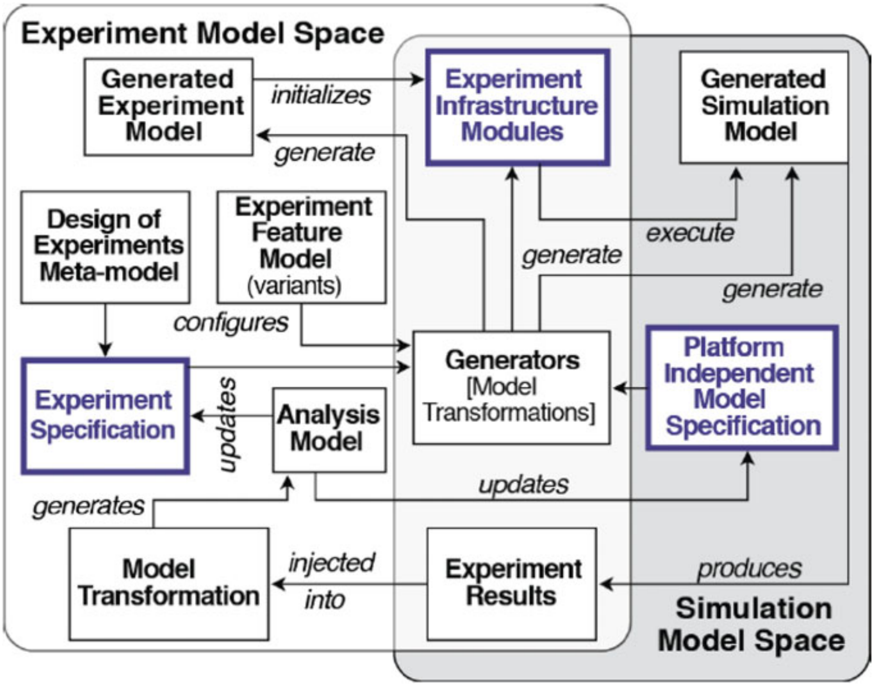


Fig. 9.9 A component architecture for experiment and model technical spaces

simulations across a variety of platforms. While the utility of MDE principles in simulation development is now widely recognized, its benefits for experimentation have not yet received sufficient attention.

In (Yilmaz 2016), a conceptual framework is presented to integrate MDE, agent models, and product-line engineering to manage the overall lifecycle of a simulation experiment. Building on this framework, the major elements of the proposed strategy are shown in the component architecture presented in Fig. 9.9. In the component architecture, the experiment and simulation model spaces are tightly coupled to orchestrate the co-evolution of simulation and experiment spaces as learning takes place. Next, we overview these components to elucidate their potential contributions to the process of computational discovery.

9.5.1 DSLs for Experiment and Hypothesis Modeling

For generating experiment specifications from research questions and hypotheses, the DOE methodology in simulation experiment design (Kleijnen 2008) provides a structured basis for automation. The DOE ontology defines the vocabulary and grammar, i.e., the abstract syntax for building the experiment domain model.

To support the instantiation of the experiment specifications conforming to the DOE meta-model, a suitable Domain-Specific Language (DSL) (Teran-Somohano et al. 2015; Yilmaz et al. 2016) is needed. The experiment model defined by the DSL (Visser 2007) needs to be configured with the aspects specified in an experiment feature model. An experiment design can have various mandatory, alternative, and optional features, which are the salient attributes that facilitate modeling variants of experiments to support different objectives (Sanchez 2005). For instance, the type of the experiment design (e.g., factorial, fractional factorial), the optimization strategy (e.g., evolutionary strategy vs. simulated annealing), and the analysis method (e.g., ANOVA vs. MANOVA) are potential features that collectively define plausible configurations of an experiment.

9.5.2 Agent-Assisted Experiment Specification Generation

An experiment design agent evaluates questions of interest to generate an experiment design that is not only effective in discriminating rival hypotheses, but also efficient in covering the parameter space of the system. A trade-off analysis between the number of design points and the number of replicates per design point are carried out in relation to the type of experiment being conducted. Consider, for instance, two options: one with many replicates per design point, and another with more design points with fewer replicates. The first option enables explicit estimation of response variances that can vary across scenarios. If the primary objective is to find a robust system design, then some replication at every design point is essential. If the goal is to understand the system behavior, this requires understanding the variance, again mandating replication. However, if the goal is that of comparing systems and a constant variance can be assumed, then this constant can be estimated using classic ordinary least squares regression. Replication is then of less concern, and the second option (exploring more design points/scenarios) can be a better way to spend limited computer resources.

9.5.3 Agent-Monitored Experiment Orchestration and Update

Experiment orchestration involves experiment design adaptation capabilities so that factors that are not significant in explaining the differences in the dependent variables are reclassified as control variables, and, if necessary, design schema can adapt as experimentation moves from variable screening to factor analysis. The aggregation of results for effective analysis and communication is a critical step. Regression trees and Bayesian networks are effective ways of communicating which factors are most influential on the performance measures.

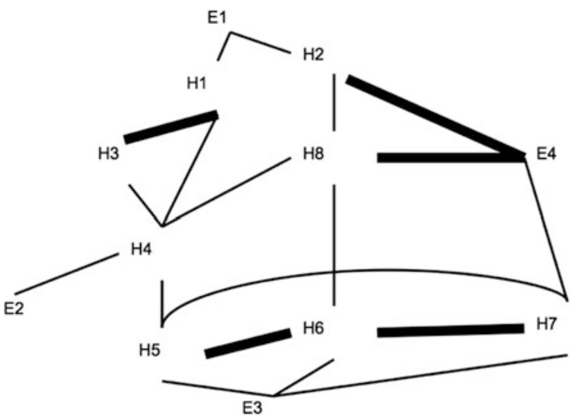
Model updating based on the analyses performed by an Experiment Orchestration Agent is the next step. We consider two types of updates: (1) experiment model (space) update and (2) simulation model (mechanistic hypothesis space) update. Adaptation of an experiment occurs at multiple levels. Based on sequential experiment results, specific factors are identified as significant, while others are classified as control variables. The reduction in the number of pertinent factors triggers a more detailed analysis of the levels of relevant factors. Such changes in the direction of exploration of the parameter space do not require an update in the experiment schema. However, higher order experiment schema (meta-model) and search strategy adaptation may be necessary when the observed response surface complexity and the change in the number of factors trigger, for example, an update from a Central Composite Design to a Latin Hypercube design. Schema adaptation can be followed by a complete schema revision, requiring a new experiment model consistent with the evolving focus of the experiment.

9.6 Cognitive Computing as an Aid to Computational Discovery

To support experimentation within the experiment and hypothesis spaces, there is a need to conjecture plausible explanations for the targeted behavior and discern a coherent set of mechanisms that collectively work together to generate it. To this end, we put forward the use of explanatory coherence theory (Thagard 1989) as an illustrative example shown in Fig. 9.10. The proposed implementation technique uses a self-organizing coherence maximization approach based on the Interactive Activation Competition model (McClelland and Rumelhart 1989) to discern the combination of mechanisms that fit together to exhibit the desired behavior.

To incorporate the theory of explanatory coherence into this framework, one can define a DSL with features that represent the hypotheses and evidences, along with

Fig. 9.10 Hypothetical coherence network



initial facilitation and inhibition relations among them. These relations are subject to change based on the results of experiments. The empirical evidence or expected behavior can be presented in terms of finite state verification patterns, which are compiled into Linear Temporal Logic. Using a probabilistic model checker such as PRISM (Kwiatkowska et al. 2002), scientists can examine if the evidence is supported by the mechanisms of the model.

The theory of Cognitive Coherence builds on the notion of establishing relations among propositions. Coherence between two propositions is achieved if any of the following is true: (1) P is part of the explanation of Q. (2) Q is part of the explanation of P. (3) P and Q are together part of the explanation of some R. (4) P and Q are analogous in the explanations they, respectively, give of some R and S. For illustrative purposes, in Fig. 9.11, we present a hypothetical coherence network that is comprised of evidence and hypothesis nodes. In this example, hypotheses *H1* and *H2* together explain the evidence *E1*. The evidence can be represented by a predicate or expected pattern, whereas hypotheses are the behavioral mechanisms that when enacted generate model behavior consistent with the evidence. That is, *H1* and *H2* explain the evidence *E1*. An edge with a solid thin line indicates facilitation or explanation relation among two nodes, whereas a thick line denotes an inhibition relation. For instance, *H8* contributes to *H4* and *H2*, whereas it inhibits *E4*. With the coherence relations at hand, synthesizing a model that is capable of and effective in explaining/generating a set of targeted behaviors can be viewed in terms of the coherence problem.

The Coherence Problem: The coherence problem is defined as follows: We define a finite set of elements e_i and two disjoint sets, $C+$ of positive constraints, and $C-$ of negative constraints, where a constraint is specified as a pair (e_i, e_j) and weight w_{ij}. The set of elements are partitioned into two sets, A (accepted) and R (rejected), and $w(A, R)$ is defined as the sum of the weights of the satisfied constraints. A satisfied constraint is defined as follows: (1) if (e_i, e_j) is in $C+$, then e_i is in A if and only if e_j is in A. (2) if (e_i, e_j) is in $C-$, then e_i is in A if and only if e_j is in R. The underlying dynamics of coherence maximization is akin to simultaneous firing of neurons. Each unit receives input from every other unit that it is connected. The inputs are then moderated by the weights of the link from which the input arrives. The activation value of a unit is updated as a function of the weighted sum of the inputs it receives. The process continues until the activation values of all the units settle by no longer changing over a pre-specified limit. More formally, if we define the activation level of each node j as a_j, where a_j ranges from -1 (rejected) and 1 (accepted), the update function for each unit is as follows:

$$a_j(t+1) = \begin{cases} a_j(t)(1-\theta) + net_j(M - a_j(t)), & \text{if } net_j > 0 \\ a_j(t)(1-\theta) + net_j(a_j(t) - m), & \text{otherwise} \end{cases}$$

The variable *theta* is a decay parameter that decrements the activation level of each unit at every cycle. In the absence of input from other units, the activation level of the unit gradually decays. In the equation, m is the minimum activation and M is the maximum activation; net_j is the net input to a unit, defined by the following equation: $\sum_{i=1}^{n} w_{ij}a_i(t)$. These computations are carried out for every unit until the network reaches equilibrium. Nodes with positive activation levels at the equilibrium state are discerned as maximally coherent propositions. For experimentation purposes, the design of the network can be calibrated and fine-tuned to alter the weights of individual links representing the significance of the constraints. Furthermore, initial activation levels of the propositions and initial levels of evidential support can be set to provide priority or higher weight to specific evidences and hypotheses.

9.7 Conclusions

In this chapter, we examined the role of computational models and simulations to steer the process of scientific discovery. Following the characterization of the nature of scientific knowledge and activities, the issues and challenges in information processing and cognitive activities involved in scientific problem solving are delineated. These challenges resulted in the formulation of a reference architecture that views models as autonomous adaptive agents with learning capabilities. The premise of the strategy is based on the observed need for mediating between the theory and data so as to facilitate their coherent integration. By aligning the mechanistic hypotheses or assumptions of a model with the empirical evidence or expected regularity, the mediation process aims to facilitate the process of computational discovery by facilitating a search process across the technical spaces of models and experiments. Such mediation requires flexibility in generating and evolving multiple models. The generative process continues until it converges to one or more competing and complementary models that can explain the systemic properties of the phenomena of interest.

To support this generative process, we suggest leveraging the Model-Driven Engineering strategy and by explicitly separating the model and experimentation technical spaces. The technical space of models adapts based on the feedback received from the results of the experiments conducted within the technical space of experiment. The modification of model mechanisms is guided by a dual search process, where the search in the modeling space results in alternative hypothetical problem representations. These representations are translated into simulation code via model transformations, resulting in concrete and executable simulation models. The search in the experiment space generates experiment designs, which are orchestrated by intelligent agents to determine the explanatory power of the mechanistic hypotheses implemented in the simulation model. We also put forward the use of cognitive models in guiding the search process to provide an explanatory framework underlying the decisions made by the discovery system. Both the Model-Driven Engineering and the cognitive computing facets of the proposed reference architecture outlines a research roadmap for addressing the issues and challenges in accelerating scientific discovery.

Review Questions

1. What is the role of cognitive computing in accelerating simulation-based scientific discovery within the life cycle of a modeling and simulation study?
2. Given that models are idealized representations of a scientific phenomenon, how can scientists determine which aspects of models make trustworthy predictions or can reliably be used in explanations? This problem is confounded especially when scientists are confronted with uncertainty, ambiguity, and lack of accepted theories that can guide the strategies for model building.
3. How can model development and experimentation be coupled together to take advantage of vast amounts of data generated to improve model accuracy?
4. Which trends in computing and information sciences can be exploited to improve the activities in simulation-based scientific discovery?
5. How can we bridge the gap between the domain terminology of the scientists and the technical representations of simulation models in specific platforms so as to exploit domain models in the simulation code?

References

Blair, G., Bencomo, N., & France, R. B. (2009). Models@ run. time. *Computer*, *42*(10), 22–27.

Bosch, J., Capilla, R., & Hilliard, R. (2015). Trends in systems and software variability. *IEEE Software*, *32*(3).

Bunge, M. (1998). *Philosophy of science: from problem to theory* (Vol. 1). Transaction Publishers.

Darden, L. (2002). Strategies for discovering mechanisms: Schema instantiation, modular subassembly, forward/backward chaining. *Philosophy of Science*, *69*(S3), S354–S365.

Darema, F. (2004). Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In *International Conference on Computational Science* (pp. 662–669). Springer Berlin Heidelberg.

Davis, P. K. (2000). Dealing with complexity: exploratory analysis enabled by multiresolultion, multiperspective modeling. In *Proceedings of the 32nd conference on Winter simulation* (pp. 293–302). Society for Computer Simulation International.

Davis, P. K. (2015). Specification and Implementation of Social Science Models. In *Concepts and Methodologies for Modeling and Simulation* (pp. 275–287). Springer International Publishing.

Gelfert, Axel. "How to do science with models: a philosophical primer." *Sl: Springer* (2016).

Hey, T., Tansley, S., & Tolle, K. M. (2009). *The fourth paradigm: data-intensive scientific discovery* (Vol. 1). Redmond, WA: Microsoft research.

Honavar, V. G., Hill, M. D., & Yelick, K. (2016). Accelerating science: A computing research agenda. *arXiv preprint* arXiv:1604.02006.

Kang, K. C., Lee, J., & Donohoe, P. (2002). Feature-oriented product line engineering. *IEEE software*, *19*(4), 58.

Klahr, D., & Simon, H. A. (1999). Studies of scientific discovery: Complementary approaches and convergent findings. *Psychological Bulletin*, *125*(5), 524.

Klahr, D. (2002). *Exploring science: The cognition and development of discovery processes*. MIT press.

Kleijnen, J. P. (2008). *Design and analysis of simulation experiments* (Vol. 20). New York: Springer.

Kwiatkowska, M., Norman, G., & Parker, D. (2002, April). PRISM: Probabilistic symbolic model checker. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation* (pp. 200–204). Springer Berlin Heidelberg.

Langley, P. (2000). The computational support of scientific discovery. *International Journal of Human-Computer Studies*, *53*(3), 393–410.
McClelland, J. L., & Rumelhart, D. E. (1989). *Explorations in parallel distributed processing: A handbook of models, programs, and exercises*. MIT press.
Morrison, M., & Morgan, M. S. (1999). Models as mediating instruments. *Ideas in context*, *52*, 10–37.
Mosterman, P. J. (1999). An overview of hybrid simulation phenomena and their support by simulation packages. In *International Workshop on Hybrid Systems: Computation and Control* (pp. 165–177). Springer Berlin Heidelberg.
Oliveira, B. C. D. S., Van Der Storm, T., Loh, A., & Cook, W. R. (2013). Feature-Oriented programming with object algebras. In *European Conference on Object-Oriented Programming* (pp. 27–51). Springer Berlin Heidelberg.
Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.
Sanchez, S. M. (2005). Work smarter, not harder: guidelines for designing simulation experiments. In *Proceedings of the 37th conference on Winter simulation* (pp. 69–82). Winter Simulation Conference.
Teran-Somohano, A., Smith, A. E., Ledet, J., Yilmaz, L., & Oğuztüzün, H. (2015). A model-driven engineering approach to simulation experiment design and execution. In *Proceedings of the 2015 Winter Simulation Conference* (pp. 2632–2643). IEEE Press.
Thagard, P. (1989). Explanatory coherence. *Behavioral and brain sciences*, *12*(03), 435–467.
Visser, E. (2007). Domain-specific language engineering. *Proceedings of the Summer School on Generative and Transformational Techniques in Software Engineering (GTTSE'07), lcns. Springer Verlag*.
Völter, M., Stahl, T., Bettin, J., Haase, A., & Helsen, S. (2013). *Model-driven software development: technology, engineering, management*. John Wiley & Sons.
Yilmaz, L., & Oren, T. I. (2004). Dynamic model updating in simulation with multimodels: A taxonomy and a generic agent-based architecture. *Simulation Series*, *36*(4), 3.
Yilmaz, L., Lim, A., Bowen, S., & Oren, T. (2007, December). Requirements and design principles for multisimulation with multiresolution, multistage multimodels. In *2007 Winter Simulation Conference* (pp. 823–832). IEEE.
Yilmaz, L. (2016, July). Higher order synergies among agents, simulation, and model-driven engineering. In *Proceedings of the Summer Computer Simulation Conference* (p. 4). Society for Computer Simulation International.
Yilmaz, L., Chakladar, S., & Doud, K. (2016). The Goal-Hypothesis-Experiment Framework: A Generative Cognitive Domain Architecture for Simulation Experiment Management, in Proceedings of the *2016 IEEE/ACM Winter Simulation Conference*, pp. 1001–1012. December 11–14, 2016, Washington, D.C.

Author Biography

Levent Yilmaz is Professor of Computer Science and Software Engineering at Auburn University with a joint appointment in Industrial and Systems Engineering. He holds M.S. and Ph.D. degrees in Computer Science from Virginia Tech. His research interests are in agent-directed simulation, cognitive computing, and model-driven science and engineering. He is the founding organizer of the Agent-Directed Simulation Conference series and the former Editor-in-Chief of the Simulation: Transactions of the Society for Modeling and Simulation International.

Chapter 10
Systems Design, Modeling, and Simulation in Medicine

Hannes Prescher, Allan J. Hamilton and Jerzy W. Rozenblit

Abstract Health care is changing at a very rapid pace. So does its attendant complexity and ever increasing reliance on high technology support. Technical medicine, where sophisticated, technology-based methods are used in education of healthcare professionals and in treatment of patients, is becoming a recognized discipline. Such methods require a new generation of engineers, scientists, systems designers, and physicians to integrate medical and technical domains. With this in mind, this chapter provides an overview of modeling and simulation technologies as applied to healthcare. A historical perspective is given followed by the discussion of how simulation helps in gaining professional competency and how it improves healthcare outcomes. Systems for support of medical training and clinical practice are discussed from both engineering and clinical perspectives. Challenges and opportunities for further development of complex simulation-based medical trainers are presented as well.

Keywords Future developments in medical simulation · History of simulation-based medical education · Simulation for clinical training · Simulation for healthcare · Simulation to evaluate healthcare outcomes · Simulation to improve healthcare outcomes · Simulation-based medical education

10.1 Introduction

Modeling and Simulation (M&S) is a mature scientific and engineering discipline, where rigorous, theory-based foundations (Zeigler 1976) gained considerable footing. The field spans a broad spectrum of contexts, e.g., mathematics, natural systems in physics (computational physics), astrophysics, chemistry and biology, economics, psychology, social science, engineering, and now, healthcare. In the last

H. Prescher · A.J. Hamilton · J.W. Rozenblit (✉)
Department of Electrical and Computer Engineering, University of Arizona,
1230 E Speedway Boulevard, 85721-0104 Tucson, AZ, USA
e-mail: jr@ece.arizona.edu

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_10

decade or so, we have witnessed burgeoning interest in and demand for simulation-based education and training in healthcare fields (Rozenblit and Sametinger 2015). The motivation and rationale are clear: through the use of simulation in medical training, a safer patient experience will result by preventing medical errors and by improving outcomes. The benefits of such an approach are manifold: (a) healthcare providers can practice procedures, techniques and responses to various scenarios without any risk to patients. Such exercises are infinitely repeatable, (b) training and education can occur in true-to-life environments, with facilities and technology identical to those used in various medical settings, (c) all learners—expert physicians to high school students—can benefit from simulated experiences, and (d) training can support the development of a wide variety of skills without the risk to patients and sacrifice of animals.

In this chapter, we give an overview of the history and current uses of simulation in healthcare. We also address the methodological challenges for development of techniques, validation, and design of features that can leverage from the rigorous science of modeling.

10.2 History of Simulation-Based Medical Education (SBME) and Its Current Use in Undergraduate Medical Education (UME)

The use of simulation in medical education traces its roots to the origins of modern medicine. In the 17th century, mannequins, referred to at the time as phantoms, were used to teach obstetrical skills. Cadavers have long been used along with clay and wax models to teach human anatomy (Owen 2012). Several developments in the middle of the 20th century, both technological and ideological, transformed medical simulation into its current form (Bradley 2006).

10.2.1 Simulation Environments

The first was the technological improvement of part-task trainers. In 1960, Asmund Laerdal launched “Resusci-Anne,” a part-task trainer for teaching cardiopulmonary resuscitation (CPR) to emergency medical technicians (Cooper and Taqueti 2004). The trainer revolutionized resuscitation training through widespread availability of a low-cost, effective training model (Lind 2007; Grenvik and Schaefer 2004). Other, more technologically advanced part-task trainers followed (Issenberg 2005). The most notable of these was the Harvey mannequin, a cardiopulmonary patient simulator launched in 1968 at the University of Miami that was designed to mimic the basic cardiac functions of the human body (Gordon 1974; Gordon et al. 1980). Using a series of cams and levers to create heart movement and a 4-track tape recording for sound, the trainer allowed students to practice cardiac auscultation skills. In so doing, it became the first trainer to provide a standardized method of testing bedside cardiovascular examination skills (Cooper and Taqueti 2004).

The second development was the introduction in the late 1960s of Sim One, the first computer-driven high-fidelity patient simulator capable of reproducing physiological functions of the entire patient (Abrahamson et al. 1969, 2004). The mannequin was controlled by a hybrid digital and analog computer and was designed for anesthesiologists to practice endotracheal intubation. However, Sim One failed to achieve acceptance as a training model, due in part to its divergence from the widely accepted apprenticeship model of medical education at the time (Bradley 2006). Advances in the 1980s of mathematical models of human physiology along with marked acceleration in computing power led to the development of the high-fidelity patient simulators widely used today (Owen 2012).

Building on the concept of screen-based simulators capable of running on a desktop computer, a new human patient simulator, the Comprehensive Anesthesia Simulation Environment (CASE), was designed at Stanford University and combined commercially available waveform generators on a desktop computer with a commercially available mannequin (Cooper and Taqueti 2004; Gaba and DeAnda 1988). This mannequin, whose vital signs could be adjusted to create different clinical events, was placed in a real operating room and was used with the express intent to improve patient safety under anesthesia through team-based training (Gaba et al. 2001). A training curriculum was designed based on the aviation model of crew resource management and thus emerged a program of performance assessment of both technical and behavioral skills in medical education (Grenvik and Schaefer 2004).

The third and most significant development was the movement in the late 20th century of medical education reform (Bradley 2006). For over a century, the undergraduate medical curriculum rested primarily on intense didactic learning coupled with an apprenticeship model of clinical observation. This led to information overload at the expense of learning clinical and team-based skills and produced medical students that were ill equipped to face the demands of an increasingly complex healthcare system (General Medical Council 1993; Cartwright et al. 2005; Feher et al. 1991). At the same time, a landmark Institute of Medicine report *To Err is Human* exposed institutional deficiencies in patient safety and created an ethical imperative to promote the training of complex technical and behavioral skills in a setting that does not compromise patient care (Kohn et al. 2002; Ziv et al. 2003, 2005). These forces combined to promote the widespread adoption of SBME.

The last decade has seen advances in simulation technologies that continue to improve the fidelity of simulation environments (Bradley 2006; Cooper and Taqueti 2004). Today, the scope of medical simulation modalities available ranges from low-tech models used to practice simple physical maneuvers or procedures to realistic computer-driven patient simulators that simulate the anatomy and physiology of real patients and provide learners with an immersive environment in which to practice complex, high-risk clinical situations in a team-based setting.

Screen-based computer simulators have been developed to train and assess clinical decision-making (Schwid et al. 2001; Bonnetain et al. 2010). Complex task trainers including virtual reality simulators provide fully immersive, high-fidelity visual, audio, and touch cues along with actual tools integrated with computers to replicate a clinical setting (Cook et al. 2010). These complex simulators allow students to develop technical skills in ultrasound, bronchoscopy, laparoscopic surgery, arthroscopy, and cardiology (Khanduja et al. 2016; Konge et al. 2011; Beyer-Berjot et al. 2016). Despite the recent advances in simulation technologies, the most common simulation modality used in UME continues to be the standardized patient—actors trained to role-play patients for training history taking, physical examination and communication skills (Keifenheim et al. 2015).

Standardized patients have played an integral part in the most established simulation exercise in UME, namely the Objective Structured Clinical Examination (OSCE) (Newble 2004). First described in 1979, the OSCE was designed to assess the clinical competency of medical students by using clinical scenarios with patient actors to test communication and professionalism, history taking, physical examination, and clinical reasoning skills (Harden and Gleeson 1979). These simulated patient encounters have become a required part of the United States Medical Licensing Examination (USMLE) and a strong emphasis is therefore placed on practicing these encounters. While standardized patients are useful in learning basic clinical skills, they do not challenge the learners to train in the types of interdisciplinary teams required to provide efficient care coordination (Patricio et al. 2013). They also do not allow learners to perform invasive procedures and often fail to provide accurate diagnostic cues.

Other simulation modalities have gained appeal as their utility has become clear. In a survey conducted by the Association of American Medical Colleges (AAMC) in 2011 on the use of medical simulation in medical education, 95% of respondents reported using full-scale mannequins, while 93% reported using partial task trainers to train students in clinical skills, clinical medicine and physical diagnosis (Passiment et al. 2011). Using the six core competencies set by the Accreditation Council for Graduate Medical Education (ACGME) of medical knowledge, patient care, interpersonal communication skills, professionalism, practice-based learning, and system-based practice—competencies that undergraduate medical students are expected to satisfy—86% of respondents reported using some form of simulation to train students in these competencies. However, only 46% of this training was done in multidisciplinary, interprofessional teams (Cook et al. 2010). Likewise, there was large inconsistency in the types of partial task trainers used with most medical schools only providing basic trainers for suturing, IV access, and airway management.

10.2.2 University Education

Undergraduate SBME is supported by a strong theoretical foundation. Proponents of SBME emphasize the ability for repetitive practice of procedural skills in a safe, controlled environment, and the ability to personalize the training to the needs of individual learners and capture clinical variation to standardize medical training (Moorthy et al. 2005). It provides the opportunity to give immediate feedback to learners, and creates an environment for competency-based mastery-learning based on defined outcomes (Deutsch et al. 2016). A systematic review and meta-analysis of 609 studies was conducted to assess the effectiveness of technology-enhanced simulation for health professions education (Cook et al. 2012). It revealed that, in comparison with no intervention, technology-enhanced simulation is associated with large effects for outcomes of knowledge, skills, and behaviors and moderate effects for patient-related outcomes. When compared to other instructional methods (i.e., lectures, small group discussions), technology-enhanced simulation was associated with small to moderate positive effects on learning outcomes.

The authors noted significant inconsistencies in effect size between studies but were able to isolate several components of simulation interventions that were consistently associated with significantly higher outcomes. Higher feedback and learning time, group work and lower cognitive load all contributed to higher effect size in simulation versus the comparison intervention (Cook et al. 2013a). Similarly, interventions using a mastery-learning model that requires learners to achieve a benchmark of skills before proceeding to higher level exercises were associated with significantly better outcomes, as were those with curricular integration (Cook et al. 2011; McGaghie et al. 2011a, b).

These studies suggest that the merits of simulation may vary for different educational objectives. Designing undergraduate medical curricula to align educational objectives with instructional modalities, therefore, will be critical to maximize the effectiveness of the intervention and justify the financial investment. The literature does provide empirical evidence for the importance of creating an integrated, longitudinal simulation program that is aligned with physiology, pathology and pharmacology topics learned in the didactic setting (Gorman et al. 2015; Gordon et al. 2006; Rosen et al. 2009). Using this model, SBME provides an opportunity for active, team-based collaborative learning that promotes transfer of basic clinical knowledge to treat real clinical problems. Simulated patient encounters, whether in a virtual reality environment with avatars or using a computerized high-fidelity mannequin should challenge learners to apply their knowledge in a simulated clinical setting with the type of emotional engagement and stress encountered in the real clinical setting (Hunt et al. 2007).

As medical schools reshape their curricula to incorporate simulation training, it will be important to continue to study the impact of elements of instructional design (i.e., learning time, repetition, feedback) versus the effect of modality (i.e., simulation tools) on learning outcomes to maximize the efficiency of resource utilization. Figure 10.1 depicts the key stakeholders in healthcare simulation and summarizes its benefits.

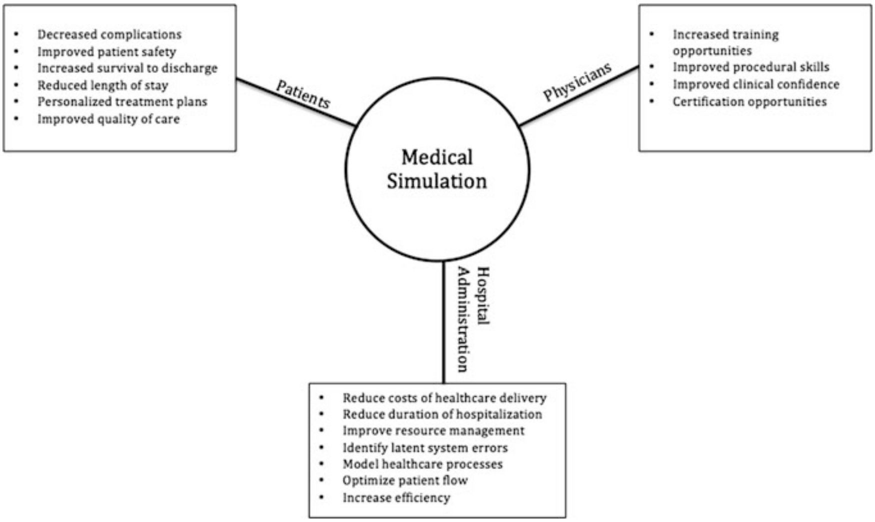


Fig. 10.1 Healthcare simulation stakeholders and benefits

10.3 Use of Simulation for Clinical Training and Acquisition of Procedural Competency

The traditional model for procedural training in medicine was established by William S. Halsted in 1890 when he transformed graduate medical education by creating the first surgical residency program at John Hopkins University (Kotsis and Chung 2013). Based on the principle of graduated responsibility, Halsted used the adage "see one, do one, teach one" as a pedagogical framework for acquiring procedural competency. This system of learning remained unchanged for over a century. Reductions in resident duty hours instituted by the Accreditation Council for Graduate Medical Education (ACGME) in 2003 along with a national movement to improve patient safety finally led to a paradigm shift (Rodriguez-Paz et al. 2009). At the same time, simulation-based medical education (SBME) emerged as an alternative training model that enabled learners to safely acquire procedural competency without causing harm to patients. A new pedagogical framework was proposed that integrates both the cognitive and psychomotor phases of learning with deliberate practice and emphasizes formative and summative assessment with defined benchmarks for skill acquisition (Sawyer et al. 2015).

Using simulation for procedural training has gained acceptance in many graduate medical education programs. The ACGME requires simulation-based training opportunities for trainees in anesthesiology, general surgery and internal medicine and accepts it as a method of training, assessment and evaluation in emergency medicine, ophthalmology, otolaryngology, radiology, and urology (Deutsch et al. 2016). The American Board of Surgery requires successful completion of competency-based skills training in simulation to achieve eligibility for board certification and the American Board of Anesthesiology and the American Board of Internal Medicine permit the use of simulation for maintenance of certification in required procedural skills (Deutsch et al. 2016).

As acceptance of simulation-based training has grown within an increasingly complex healthcare system, a new generation of engineers, scientists, systems designers and physicians has been challenged to develop new technologically complex training models to replicate partial and complete organ systems for practicing procedural skills. Using a variety of synthetic materials (i.e., plastics, silicones etc.), biomaterials and fluids, bioengineers and clinical researchers have developed a vast collection of life-like, anatomically accurate procedural trainers. Medical trainees now routinely use artificial arms to practice peripheral IV and arterial line placement. Models have been designed to practice, amongst other procedures, endotracheal intubation, cricothyrotomy, lumbar puncture, thoracentesis, thoracostomy as well as central line and urethral catheter placement (Nestel et al. 2011).

Surgery is perhaps the medical specialty most reliant on psychomotor and visuospatial skills and routinely performs procedures that pose significant risks of patient morbidity. This makes it uniquely suited for simulation training and it has been the specialty, along with Anesthesiology, that has been historically most invested in it. Surgical trainees have long used silicone pads to practice basic suturing and knot-tying techniques and various-sized silicone tubing to practice open and microsurgical anastomosis of blood vessels (Badash et al. 2016). In an effort to standardize basic surgical skills training, the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) in 2004 launched Fundamentals of Laparoscopic Surgery (FLS), a formal simulation-based education program for teaching the basic skills of laparoscopic surgery (Zendejas et al. 2016). Using a series of manual tasks within a simple box trainer to simulate the surgical working space, learners were able to train in the critical skills of depth perception, spatial orientation, and manual dexterity that form the foundation for safely performing laparoscopic surgery (Scott 2006). The educational program is based on the principles of mastery-learning; trainees are required to complete the series of tasks within validated time and performance benchmarks to prove procedural competency (Schout et al. 2010).

Further technological advances in surgical simulation have seen the emergence of computer-driven computational models of whole organ systems to create virtual reality trainers (Aggarwal et al. 2007). These simulators allow users to interact with a virtual image using a physical interface that is identical to the actual instrument used in the clinical procedure (Deutsch et al. 2016). For instance, in a virtual cholecystectomy, the user holds instruments shaped like a retractor, clip-applier and scissor to manipulate, ligate and cut the appropriate virtual anatomy (Schijven et al. 2005). The simulators provide haptic feedback to learners to give the tactile sensation of actually performing the procedure. They are also sensitive to the forces applied to virtual anatomic structures by the user and can simulate surgical complications such as bleeding. Similar simulators exist for practicing bronchoscopy, arthroscopy, cardiac catheterization and other complex procedures (Gallagher and Cates 2004).

The unique nature of virtual reality trainers lies in their ability to actively engage the learners' senses and provide the most immersive learning experience available (Alaker et al. 2016). Adult learning theory in medicine dictates that multisensory engagement in a learning activity is essential for effective learning and retention of skills (Kneebone 2005; Rutherford-Hemming 2012). The degree of immersion in current virtual reality procedural trainers operates on a spectrum from simple interactive gaming platforms (e.g., mobile and desktop applications) to comprehensive 3D surgical environments. Despite proven effectiveness in facilitating basic surgical skill acquisition, virtual reality technology in medicine remains in its infancy (Larsen et al. 2012). Recent advances in CAD and 3D printing technologies promise to expand our ability to create patient-specific 3D models in a virtual simulation environment to aid surgeons in evaluating and simulating pre-operative treatment options.

As medical education evolves towards a personalized, competency-based training system, the role of assessment will continue to grow (Michelson and Manning 2008). In order to define procedural competency and prove mastery of a skill in a simulated environment, training programs have developed and validated procedure-specific assessment tools. These tools include global rating scales and checklists (Ilgen et al. 2015; Riojas et al. 2011). Other assessment tools, including FLS in surgery training, use objective variables such as task completion time, the quality of a finished product (e.g., accuracy of stitch placement in a suturing task) or procedural errors as measures of proficiency (Feldman et al. 2004). Newer computer-based trainers are able to provide a performance score based on the users' economy of instrument movement and overall efficiency in completing a task (Chang et al. 2016). However, a recent systematic review of validity evidence of commonly used assessment tools reports that the methodological and reporting quality of validity testing is inconsistent and often inadequate (Cook et al. 2013b). If the role of simulation is to continue to grow and influence decisions regarding trainee preparedness, remediation and credentialing, assessment tools will need to be subjected to rigorous validation testing to support the interpretation of scores.

A great body of evidence has emerged over the last decade in simulation research in support of the effectiveness of various simulated learning platforms (Cook et al. 2013a). Studies have shown that deliberate practice in a simulation environment can improve procedural skills compared to traditional methods of training. However, a major challenge in simulation training and a persistent point of contention among clinicians is providing evidence that skills acquired on a simulated model transfer to the clinical setting. For instance, does achieving proficiency in surgical skills in a box trainer translate to acceptable skill levels in the operating room? Studies across specialties using a diverse set of simulation platforms to learn a large variety of procedures have provided evidence that practicing a procedure using a mastery-learning model leads to superior performance of that procedure in a clinical setting (Huang et al. 2016; Brydges et al. 2015; McGaghie et al. 2010). Furthermore, it has been shown that repeated practice of procedural skills facilitates the retention of skills over time (Shewokis et al. 2016; Opoku-Anane et al. 2015; Sant'Ana et al. 2016).

Restrictions in resident duty hours were implemented by the ACGME to improve resident fatigue and burnout and increase patient safety. However, in the process it has also inadvertently decreased the learning opportunities of residents especially in clinical procedures. Increased clinical demands of faculty in a profit-driven healthcare system have also limited the time available for formal instruction and mentorship during patient care. In this setting, SBME offers a viable alternative training model that is congruent with efforts to improve patient safety by providing the opportunity for deliberate practice on procedural simulators without compromising resident duty hours.

10.4 Use of Simulation to Evaluate and Improve Healthcare Outcomes

Healthcare organizations are complex ecosystems composed of integrated networks of human teams supported by technological systems. These organizations are challenged by political, social and economic forces to improve the value of healthcare delivery by providing safer, higher quality care at lower costs (McGinnis et al. 2013). Complicating this effort are inefficiencies in the *individual*, *team* and *systemic* processes that contribute to the delivery of care. Research has consistently shown that failures in healthcare processes are multi-factorial in nature and occur because of unpredictable combinations of component failures (Kohn et al. 2002; Marshall et al. 2016). Simulation can be used as (a) dynamic prototyping platforms to better understand how *individual*, *team* and *system processes* interact, (b) methodology to assess collective performance to optimize healthcare outcomes (Deutsch et al. 2016; Isern and Moreno 2016); and (c) a tool to evaluate human performance factors and the impact of new methods or technologies.

Improving *individual* performance in healthcare organizations has been the main focus of simulation-based medical education (SBME) to date. Careful review of patient safety and quality control management databases revealed that iatrogenic human error in clinical procedures is a common source of patient morbidity and mortality (Kohn et al. 2002; Hripcsak et al. 2003). SBME programs were developed to improve individual performance in common procedural skills. Outcome studies were subsequently designed to assess the impact of the simulation intervention through pre/post analysis of procedural complications, patient survival to discharge and duration of hospitalization (Zendejas et al. 2013; Griswold-Theodorson et al. 2015; McGaghie et al. 2011a, b).

According to the Center for Disease Control (CDC), in 2009 an estimated 18,000 central line-associated bloodstream infections (CLABSIs) occurred among patients hospitalized in intensive care units (ICU) in the United States, each carrying an attributable mortality risk of 12–25% in addition to millions in excess healthcare costs (Centers for Disease Control and Prevention 2011; Klevens et al. 2007). In an effort to reduce the complication rates, a team at Northwestern University developed a simulation-based mastery-learning program for central venous catheter (CVC) placement (Barsuk et al. 2009a). Following the implementation of its simulation-based training program, the team reported an 85% reduction in CLABSIs (Barsuk et al. 2009b). Residents trained in the program experienced fewer complications including fewer needle passes, arterial punctures, catheter adjustments and higher success rates in CVC insertions in the medical ICU compared with traditionally trained residents (Barsuk et al. 2009c). The study was replicated at a second institution, where a 74% reduction in CLABSI rates was reported (Barsuk et al. 2014). The researchers further demonstrated that the improved patient outcomes resulted in significant medical care cost savings with a 7–1 return on investment (Cohen et al. 2010). In a separate study, training of critical care nurses in sterile technique using a simulation-based training program resulted in a reduction in the rate of CLABSIs of 85% in an ICU setting (Gerolemou et al. 2014).

Similar outcomes studies have been conducted in surgical simulation training (Seymour 2008). Training on a virtual reality (VR) simulator has been shown to significantly improve the performance of surgical residents in actual cholecystectomies in the operating room (Ahlberg et al. 2007; Beyer et al. 2011). In addition, residents trained on a VR colonoscopy simulator similarly performed significantly better on real patients and experienced fewer procedural complications than the control group (Park et al. 2007; Sedlack and Kolars 2004; Cohen et al. 2006). Training Ophthalmology residents in a structured surgical curriculum in cataract surgery resulted in a significant reduction in the sentinel complication rate, as defined by posterior lens capsule tear, and vitreous loss during actual surgery, from 7.17 to 3.77% (Rogers et al. 2009). Experiential simulation training in thoracentesis resulted in a decrease from 8.6 to 1.1% in the rate of pneumothorax (Duncan et al. 2009).

Strategic SBME interventions have also been designed to improve the performance of healthcare *teams* in response to evidence that poor communication among team members is a common source of avoidable medical errors. Studies in obstetrics and gynecology have demonstrated that implementation of a hospital-wide multidisciplinary simulation-based teamwork training can significantly decrease the adverse outcomes index (Phipps et al. 2012; Riley et al. 2011). In addition, targeted teamwork training in births complicated by shoulder dystocia decreased birth complication rates of brachial plexus injury and neonatal hypoxic ischemic encephalopathy and increased the APGAR scores of neonates at 5 min after birth (Draycott et al. 2008).

Simulation-based teamwork training in the actual clinical setting has also been shown to improve early trauma care and increase patient survival in cardiopulmonary resuscitation codes (Morey et al. 2002; Steinemann et al. 2011; Capella et al. 2010). Human factors experts argue that conducting simulations in the actual unit where patient care is delivered probes for overt and latent problems in the way the clinical environment influences human performance (Deutsch et al. 2016).

This can therefore be a particularly useful method to identify *system* factors that can impact patient outcomes. However, full-scale simulations in actual clinical spaces can be challenging on several levels and can be disruptive to patient.

Simulation methodology can also be applied at the system level to improve unit efficiency. Dynamic simulation modeling (DSM) is an alternative simulation platform that can be used to create computer-based representations of real healthcare processes to explore their interaction in a modeled clinical setting. The individual healthcare processes are variables that can be adjusted to see how particular changes affect the system outcomes predicted by the model (Deutsch et al. 2016; Isern and Moreno 2016; Pennathur et al. 2010). A common DSM method used in modeling healthcare systems is discrete event simulation (DES), which models the operation of a system as a discrete sequence of events in time. It is of particular value in studying resource management in a clinical setting to achieve a desired outcome, for instance to reduce the wait time for patients in the emergency room (Day et al. 2012). In this scenario, a DES model was constructed using as adjustable time-sensitive variables, triage and registration time, availability of ED beds, rooms, labs and radiology services, nurses and physicians. The model predicted that wait time was most dependent on the availability of ED beds, nurses, physicians and labs and radiology resources. Researchers subsequently adjusted these variables to identify the time-limiting resource and determined that increasing physician and mid-lever provider coverage at triage significantly reduced ED length of stay. Shorter length of stay has been directly associated with reduction in in-hospital complications and improved patient outcomes (Rotter et al. 2012).

Similar computational models have been used by healthcare managers and administrators to estimate bed capacity in an intensive care unit (ICU) setting (Zhu et al. 2012; Ferraro et al. 2015), predict staffing needs based on patient mix, patient acuity and resource costs (DeRienzo et al. 2016; Hoot et al. 2008) and optimize patient care in the cardiac ICU (Day et al. 2015). Optimizing the use of available resources allows for the delivery of more efficient, higher quality care at a lower cost.

Model parameters can be captured from a rapidly growing dataset of clinical information that includes electronic health records (EHRs), clinical research data and quality improvement data (Isern and Moreno 2016). Advances in computing have led in the past decade to the emergence of so-called "big data" as a tool for understanding health system dynamics and informing decisions about patient care delivery. Such data analysis has been defined in the bioinformatics literature as "information assets characterized by such a high *volume*, *velocity*, and *variety* as to require specific technological and analytical methods for its transformation into value" (De Mauro et al. 2015). DSM can function as the analytical method to evaluate and analyze large database to aid in the interpretation of its clinical significance and test hypotheses about the impact on patient outcomes of potential healthcare interventions (Huang et al. 2015).

A team of health economists, software engineers, data miners, business analysts, and clinicians demonstrated this system design process by integrating health informatics, activity-based costing and dynamic simulation modeling to create Network Tools for Intervention Modeling with Intelligent Simulation (NETIMIS) (Johnson et al. 2016). This tool functions as a DES model that captures the flow of individual entities (patients) through discrete events in a simulated process. The utility of the tool was demonstrated through a simulation of potential care pathways of patients presenting to the ED with sepsis. Potential care pathways included admission to critical care, admission to the ward, and discharge home, among others. The model was designed to assess the potential cost savings and reduction in adverse patient outcomes with implementation of a hypothetical point-of care testing device for early detection of severe sepsis. The sensitivity of the device in detecting early symptoms of sepsis determined initiation of respective care pathways. Patient data in the model was derived from EHRs and provided an accurate clinical model of symptoms exhibited by patients with sepsis. The tool was pre-populated with reference sets of activity-based costs so that simulated actions by healthcare providers within the model reflect current health economic cost models. The NETIMIS model demonstrates the ability of computer simulations to assess interventions within modeled clinical environments to evaluate and predict healthcare outcomes.

On a population-based level, DSM has been used synergistically with informatics to analyze chronic diseases such as diabetes, HIV/AIDS, cancer and heart disease to identify patient factors that predict their progression (Gallagher and Cates 2004; Gaba 2004). Data derived from such models have then been applied to inform decisions for patient-specific treatments (Cooper et al. 2002). For instance, physicians have used simulation models to compare various treatment options in adjuvant breast cancer therapy based on predicted outcomes and cost (Isern and Moreno 2016). Computational modeling has also found applications in predictive analytics; researchers were able to use a patient-specific DSM of epileptogenic cortexes in patients suffering from intractable, medically refractory epilepsy to determine which patients are likely to benefit from surgery (Sinha et al. 2016).

Patients interact with individual healthcare providers and healthcare teams in a rapidly changing and increasingly complex care delivery system. New knowledge, available treatments, equipment, technology, and business models constantly redefine the standard of care and challenge the optimization of care delivery. Creating an integrated, dynamic simulation architecture can help healthcare stakeholders analyze the intrinsic complexity and diversity of healthcare delivery systems and develop solutions for improving the performance of *individual*, *team-based* and *system* processes to ultimately optimize healthcare outcomes. The summary of outcomes categorized with respect to individual, team training, and systems optimization is shown in Fig. 10.2.

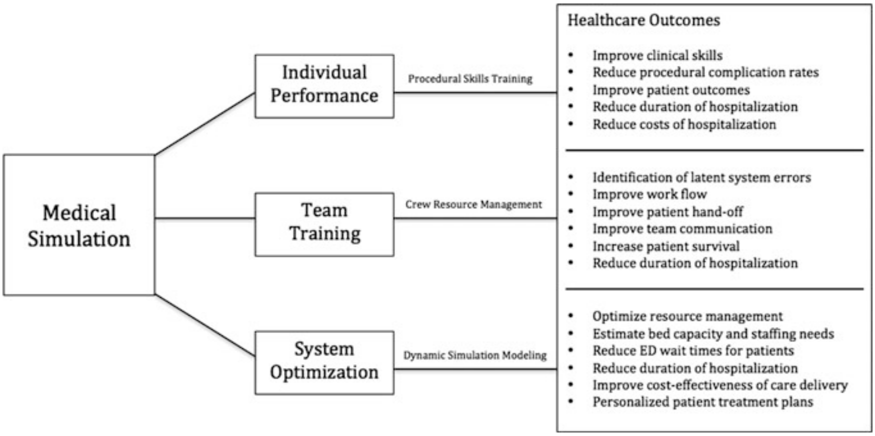


Fig. 10.2 Summary of healthcare outcomes

10.5 Design, Modeling Challenges and Opportunities

From a methodological perspective, simulation models in medicine can be classified as *simulation scenarios* and *simulation systems* (Rozenblit and Sametinger 2015). A scenario is a set of steps and actions that replicate a specific medical procedure which may be as simple as phlebotomy (making a puncture in a vain with a needle) or as complex as multiple-organ failure emergency treatment. As described above, in such cases, various actions taken by the trainees are carried out in a simulated setting, by either using actors as patients or computer-based systems and devices that emulate human anatomy and physiology. In the *systems* category, we employ models which are part of the training scenarios, and ones that are embedded in various medical devices and equipment used in actual clinical practice.

Simulation-based training cycle can be abstractly represented as shown in Fig. 10.3. Trainees are students, fellows, residents, EMS personnel, etc., who use various medical implements to carry out a training exercise/procedure. As discussed in Sects. 2–4, they can practice on low-end trainers or highly complex simulators. The low-end trainers *do not* incorporate in them the process and steps to be followed in order to perform a certain procedure—in this case, the step are typically quite simple. They are described by the supervisor or simply given through a description of the task to be completed. The high-end trainers “drive and guide” the users through a series of procedural steps. Such simulators provide feedback on how well users perform. This is done through the set of metrics. The training cycle continues until a satisfactory performance level has been achieved.

Either group, that is simulation-based training process models, and simulation systems, presents different challenges for validation, assurance of robustness, and security but also a set of exciting opportunities for leveraging from high technology in simulation-based training.

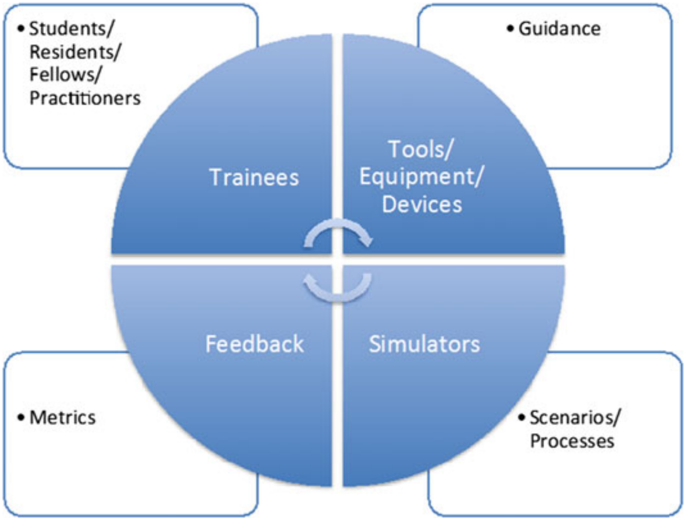


Fig. 10.3 A conceptual, simulation-based training cycle

Such training requires not only the requisite physical equipment that is used in medical procedures, but also *correct/valid* models that are the foundation for emulating the symptoms and responses to "virtual treatment". For instance, in a scenario of anaphylactic shock—a life-threatening allergic response—the model should present typical symptoms such as swelling, a weak and rapid pulse, lowered blood pressure, skin reactions, etc. The treatment by injection of epinephrine and its outcome should be reflected by the reversal of such symptoms. This is key to proper understanding of the case and learning how to appropriately treat it (Rozenblit and Sametinger 2015).

Therefore, proper validation and verification (i.e., assurance that the simulator faithfully executes the underlying models) of the trainers must be carried out prior to their deployment. In addition, as trainers become more and more sophisticated and incorporate discrete, continuous, and hybrid dynamic models, mechatronic (i.e., electronic and mechanical) devices, immersive, virtual reality environments, and very complex control software, integration of all such subsystems are non-trivial. In essence, despite the growth of new technologies for hardware and software design, networked computation, sensing, and control, the following challenges remain:

- How to tackle the complexity of the systems, which requires long design cycles, verification, and certification (if such is needed).
- How to develop unifying formalisms for specification and exploration of design options of such complex systems before that are physically built and deployed.
- How to ensure flexibility in modification and re-configuration of training systems.
- How to ensure robustness and security of medical simulation systems and devices.

In regard to the last concern, embedding models, or more specifically their realizations in the form of computational processes encoded as software and hardware, in actual medical devices presents an extraordinary set of safety, security, and reliability challenges. Such models effectively "run" complex implantable devices and are a key component in medical imaging or robotic surgeries. Consider, for example, computer assisted surgery (CAS). CAS enhances the capabilities of surgeons performing surgery (for instance, using the DaVinci surgical robot (Gettman et al. 2004) but it requires models of ultimate reliability and robustness. Indeed, it is easy to imagine the dire consequences of improper translation of the surgeon's hand's movements into a maneuver of the DaVinci's robotic arm's end-effector. This highlights the urgent need for research into assuring absolute robustness of such life-critical computing systems.

Additionally, given the recent exponential rise in cybersecurity threats and attack, security measures should be implemented to guarantee confidentiality, integrity, and availability of simulation systems. In medical simulators, confidential information includes the performance of residents during training. The integrity of information becomes critical when medical reasoning is based on this information. In simulation, modified parameters may lead to discrepancies between the simulated and the real world, thus, yielding to medical errors and declined outcomes in real patient scenarios later on (Sametinger and Rozenblit 2016).

In training scenarios with simulators there is no direct impact on real patients. However, maliciously modified simulators can have various negative consequences. For example, surgery residents may automate surgical skills based on parameters that do not exist in the real world, resulting in a negative training effect and increasing the potential for error and negative outcomes. Besides training, simulation can be used in surgery for pre-surgical planning, and for guiding or performing surgical interventions. Therefore, it is critical that the integrity and security of such models are ensured as any compromise may result in negative outcomes.

Opportunities for further research in simulation modeling for healthcare

The challenges listed above have already spurred research in realms such as space and aeronautics, industrial plants, or autonomous vehicles. Further application of the theory-based methods and techniques from the following (related) fields, to medical simulation is envisioned.

1. Design and modeling for high-autonomy systems: the intent here is to provide trainees as much assistance as possible throughout the training process so that sophisticated trainers can take on the role of a "master" in the classic "master/apprentice" model. To design such systems, modelers must integrate sensing and control features that monitor and adapt to users' performance in order to provide the "right" kind of guidance, that it assistance that is functionally correct and measured in a manner that leads to better outcomes. Design of such highly autonomous systems will clearly have features offered by artificial intelligence and machine learning.

2. Artificial Intelligence: due to the advances in computational techniques and underlying HW/SW technologies, AI is currently experiencing strong resurgence. Speech recognition, reasoning, natural language processing, planning, fuzzy logic, and heuristics all offer an exiting potential for building high-end medical simulators.

Machine Learning (ML): as pointed out in Sect. 1.3, big data, predictive analytics is already being used extensively for modeling in healthcare. In the procedural, training context, ML is envisioned as a tool to provide a user-tailored training experience, where adaptation to individual trainees will take place based on their initial skill level and degree of progress they make throughout the learning process.

10.6 Future Developments in Medical Simulation

In 2004, David M. Gaba, one of the pioneers of simulation in healthcare at Stanford University, provided a future vision of simulation in which he proposes “full integration of its applications into the routine structures and practices of healthcare” (Gaba 2004). To date, medical education continues to operate largely under an apprenticeship model of training, originally developed by Halstead in the 1920s (Evans and Schenarts 2016). Over the last decade, simulation has gained wide acceptance in graduate medical education (GME) programs as a way to train and certify physicians in particularly in procedural skills. However, the full integration of simulation into clinical education, training, and outcomes assessment has yet to occur.

One of the challenges of embracing simulation as a robust educational methodology is that its benefits only emerge after long-term implementation. Improvements in patient safety and reductions in healthcare spending, long held as the hallmark benefits of simulation training, are difficult to measure, especially when one attempts to translate those benefits into hard financial data. Without an irrefutable clinical or financial evidence base, most healthcare organizations remain reluctant to make large and significant institutional investment in simulation. This is all the more difficult as new technologies in medical simulation, like immersive virtual reality and holographic displays, for example, can be very expensive purchases until they gain a wider commercial market. Moreover, academic Medicine, like many institutional cultures can often be resistant to change. Many physicians continue to argue that training in a simulation environment cannot achieve the degree of immersion provided by actual patient care. However, several market forces are likely to force a change in the medical education model and lead to further expansion of simulation use in the future.

Reducing the cost of healthcare is likely to be the primary driving force for the wider expansion of simulation use. Medical errors and poor patient outcomes continue to be a primary cost burden for healthcare organizations in the form of wasted resources and increased length of hospital stays. Administrators and policymakers look to simulation to provide a systematic training to educate, train and assess personnel, teams and systems to provide safe clinical care. In order to improve patient safety, risk managers, insurers, and government and non-government regulatory and accrediting bodies are likely to demand a robust simulation architecture be in place (Evans and Schenarts 2016). Furthermore, reductions in work hours have led to unbalanced clinical exposure and experiences among training healthcare trainees. Simulation provides a systematic and reliable way for healthcare organizations to standardize clinical training and to train providers to specific benchmarks of competency.

As changes are made to training curricula to expand use of simulation, new informatics technology support systems must be established. Learner management systems (LMS) are playing an increasing role in tracking the progress of learners towards competency-based training goals. Tracking a learner's performance both in the simulation lab as well as the clinical setting allows for more efficient integration and smarter training. It provides educators with critical information of how best to tailor a learner's training curriculum to address specific deficiencies in clinical performance. Linking simulation training to known clinical weaknesses with high potential for error maximizes the clinical impact of the training. Using clinical performance data will inform training and allow healthcare providers to continuously refine their skills through simulation tools that will be a natural extension of the real clinical environment.

New technological support systems will facilitate the process of integration. In order to make training more realistic, use of hybrid simulators that combine standardized patients (SP) with virtual reality adjuncts are likely to grow. For example, an SP may wear an ultrasoundable skin model to enhance the diagnostic learning experience. Similarly, a simulated trauma patient may wear a synthetic chest tube model to allow learners to perform invasive procedures when clinically indicated. These VR adjuncts are flexible and adjustable allowing educators to modify the learning experience and simulate the full breadth of actual clinical encounters.

Virtual reality training environments, as a technological domain, are likely to have greater applications in simulation training (Badash et al. 2015). The degree of immersion (i.e., the sense of realism experienced by learners involved in a training task or setting) in the virtual clinical environment will steadily grow. At some time in the future, learners will be able to log into an interactive virtual environment that is a replica of the actual clinical environment. As avatars, they will interact with virtual patients and other providers as they would in the real world. Improvements in haptics technologies will eventually allow them to perform physical exams and procedural interventions on virtual patients.

Surgical simulators have made the greatest advances in this area. Using medical imaging and computer-aided design technologies, researchers have developed patient-specific VR simulators to allow surgeons to plan and practice complex procedures in a virtual environment (Vakharia et al. 2016; Makiyama et al. 2012). Moreover, 3D rapid prototyping has allowed researchers to produce accurate renditions of patient-specific anatomic variations (Endo et al. 2014). In neurosurgery, 3D printed models of patient-specific aneurysms have allowed surgeons to plan the trajectory of approach and to test different aneurysm clips for size and shape

(Kimura et al. 2009; Ryan et al. 2016). As these technologies continue to grow and the interface between the virtual and the real world continues to dissolve, surgeons will be able to harness the benefits to become more efficient and skillful in the operating room. Improving the realism of these virtual training environments will also revolutionize licensing examinations for board certification. Instead of verbally describing the steps of a surgical procedure during oral board proceedings, surgeons will be adjudged by their actual performance on virtual trainers. The ability of VR trainers to provide a standardized assessment makes this particularly effective. This type of assessment method will approach the systematic use of simulation in the aviation industry, where flight simulators have been used extensively for the purpose of certification.

In an effort to improve efficiency and reduce costs in healthcare delivery, healthcare organizations are likely to expand the use of computational modeling. To make the best use of available resources, complex simulation models will be applied to every healthcare process. Hospital inventory, staffing, scheduling, and delivery of services will be determined by models that seek to optimize patient care and satisfaction while limiting costs. Furthermore, every proposed change to the system will undergo extensive testing in a simulated environment prior to implementation. Advances in computational speed and growth of available databases will allow system engineers to construct more accurate and complete models of healthcare organizations. This will allow researchers to better isolate the impact of individual components on the operation or behavior of the system and implement changes accordingly to increase overall efficiency.

Finally, the field of health economics will make greater use of computational models to improve the delivery of individualized healthcare in clinical practice. Simulations of disease progression based on patient's individual disease states compared to a large cohort of similar patients will be used routinely to drive clinical decisions regarding care (Zafari et al. 2016). Simulation models will be used to evaluate the potential value of new products or services, as for instance telemonitoring for heart failure patients (Kolominsky-Rabas et al. 2016) or team-based care for hypertension (Dehmer et al. 2016). Prospective health outcomes will be modeled against the economic impact to create a smarter, more efficient way to deliver high-quality healthcare.

The use of simulation as a means of training healthcare providers and improving the efficiency of healthcare organizations has a big potential to grow. Progress over the last two decades has shown the great potential of simulation but more systematic, long-term implementation must be achieved to realize its true benefits—that is to create a sustainable healthcare system that produces safer, higher quality care at lower costs.

Review Questions

1. Who are the key stakeholders in medical, simulation-based training?
2. How does simulation improve professional competency?

3. How does simulation improve healthcare outcomes?
4. What are the categories of models and simulators used in healthcare training?
5. What are the technical challenges and impediments that modelers face in designing complex healthcare simulators.

References

Abrahamson, S., Denson, J. S., & Wolf, R. M. (1969). Effectiveness of a simulator in training anesthesiology residents. Academic Medicine, 44(6), 515–519.

Abrahamson, S., Denson, J. S., & Wolf, R. M. (2004). Effectiveness of a simulator in training anesthesiology residents. Quality and Safety in Health Care, 13(5), 395–397.

Aggarwal, R., Ward, J., Balasundaram, I., Sains, P., Athanasiou, T., & Darzi, A. (2007). Proving the effectiveness of virtual reality simulation for training in laparoscopic surgery. Annals of surgery, 246(5), 771–779.

Ahlberg, G., Enochsson, L., Gallagher, A. G., Hedman, L., Hogman, C., McClusky, D. A., & Arvidsson, D. (2007). Proficiency-based virtual reality training significantly reduces the error rate for residents during their first 10 laparoscopic cholecystectomies. The American journal of surgery, 193(6), 797–804.

Alaker, M., Wynn, G. R., & Arulampalam, T. (2016). Virtual reality training in laparoscopic surgery: a systematic review & meta-analysis. International Journal of Surgery, 29, 85–94.

Badash, I., Burtt, K., Solorzano, C. A., & Carey, J. N. (2016). Innovations in surgery simulation: a review of past, current and future techniques. Annals of Translational Medicine, 4(23).

Barsuk, J. H., McGaghie, W. C., Cohen, E. R., O'leary, K. J., & Wayne, D. B. (2009a). Simulation-based mastery learning reduces complications during central venous catheter insertion in a medical intensive care unit. Critical care medicine, 37(10), 2697–2701.

Barsuk, J. H., Cohen, E. R., Feinglass, J., McGaghie, W. C., & Wayne, D. B. (2009b). Use of simulation-based education to reduce catheter-related bloodstream infections. Archives of internal medicine, 169(15), 1420–1423.

Barsuk, J. H., McGaghie, W. C., Cohen, E. R., Balachandran, J. S., & Wayne, D. B. (2009c). Use of simulation based mastery learning to improve the quality of central venous catheter placement in a medical intensive care unit. Journal of Hospital Medicine, 4(7), 397–403.

Barsuk, J. H., Cohen, E. R., Potts, S., Demo, H., Gupta, S., Feinglass, J., & Wayne, D. B. (2014). Dissemination of a simulation-based mastery learning intervention reduces central line-associated bloodstream infections. BMJ quality & safety, 23(9), 749–756.

Beyer, L., De Troyer, J., Mancini, J., Bladou, F., Berdah, S. V., & Karsenty, G. (2011). Impact of laparoscopy simulator training on the technical skills of future surgeons in the operating room: a prospective study. The American Journal of Surgery, 202(3), 265–272.

Beyer-Berjot, L., Berdah, S., Hashimoto, D. A., Darzi, A., & Aggarwal, R. (2016). A Virtual Reality Training Curriculum for Laparoscopic Colorectal Surgery. Journal of Surgical Education, 73(6), 932–941.

Bonnetain, E., Boucheix, J. M., Hamet, M., & Freysz, M. (2010). Benefits of computer screen based simulation in learning cardiac arrest procedures. Medical education, 44(7), 716–722.

Bradley, P. (2006). The history of simulation in medical education and possible future direc-tions. Medical education, 40(3), 254–262.

Brydges, R., Hatala, R., Zendejas, B., Erwin, P. J., & Cook, D. A. (2015). Linking simulation-based educational assessments and patient-related outcomes: a systematic review and meta-analysis. Academic Medicine, 90(2), 246–256.

Capella, J., Smith, S., Philp, A., Putnam, T., Gilbert, C., Fry, W., & Ranson, S. (2010). Teamwork training improves the clinical care of trauma patients. Journal of surgical education, 67(6), 439–443.

Cartwright, M. S., Reynolds, P. S., Rodriguez, Z. M., Breyer, W. A., & Cruz, J. M. (2005). Lumbar puncture experience among medical school graduates: the need for formal proce-dural skills training. Medical education, 39(4), 437–437.

Centers for Disease Control and Prevention. (2011). Vital signs: central line–associated blood stream infections—United States, 2001, 2008, and 2009. Annals of Emergency Medicine, 58 (5), 447–450.

Chang, J., Banaszek, D. C., Gambrel, J., & Bardana, D. (2016). Global rating scales and motion analysis are valid proficiency metrics in virtual and benchtop knee arthroscopy simulators. Clinical Orthopaedics and Related Research®, 474(4), 956–964.

Cohen, J., Cohen, S. A., Vora, K. C., Xue, X., Burdick, J. S., Bank, S., & Greenwald, D. (2006). Multicenter, randomized, controlled trial of virtual-reality simulator training in acquisition of competency in colonoscopy. Gastrointestinal endoscopy, 64(3), 361–368.

Cohen, E. R., Feinglass, J., Barsuk, J. H., Barnard, C., O'donnell, A., McGaghie, W. C., & Wayne, D. B. (2010). Cost savings from reduced catheter-related bloodstream infection after simulation-based education for residents in a medical intensive care unit. Simulation in healthcare, 5(2), 98–102.

Cook, D. A., Erwin, P. J., & Triola, M. M. (2010). Computerized virtual patients in health professions education: a systematic review and meta-analysis. Academic Medicine, 85(10), 1589–1602.

Cook, D. A., Hatala, R., Brydges, R., Zendejas, B., Szostek, J. H., Wang, A. T., & Hamstra, S. J. (2011). Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. Jama, 306(9), 978–988.

Cook, D. A., Brydges, R., Hamstra, S. J., Zendejas, B., Szostek, J. H., Wang, A. T., & Hatala, R. (2012). Comparative effectiveness of technology-enhanced simulation versus other instructional methods: a systematic review and meta-analysis. Simulation in Healthcare, 7(5), 308–320.

Cook, D. A., Hamstra, S. J., Brydges, R., Zendejas, B., Szostek, J. H., Wang, A. T., & Hatala, R. (2013a). Comparative effectiveness of instructional design features in simulation-based education: systematic review and meta-analysis. Medical teacher, 35(1), e867-e898.

Cook, D. A., Brydges, R., Zendejas, B., Hamstra, S. J., & Hatala, R. (2013b). Technology-enhanced simulation to assess health professionals: a systematic review of validity evidence, research methods, and reporting quality. Academic Medicine, 88(6), 872–883.

Cooper, J. B., & Taqueti, V. (2004). A brief history of the development of mannequin simulators for clinical education and training. Quality and Safety in Health Care, 13(suppl 1), i11–i18.

Cooper, K., Davies, R., Roderick, P., Chase, D., & Raftery, J. (2002). The development of a simulation model of the treatment of coronary heart disease. Health care management science, 5(4), 259–267.

Day, T. E., Al-Roubaie, A. R., & Goldlust, E. J. (2012). Decreased length of stay after addition of healthcare provider in emergency department triage: a comparison between computer-simulated and real-world interventions. Emergency Medicine Journal, emermed-2012.

Day, T. E., Sarawgi, S., Perri, A., & Nicolson, S. C. (2015). Reducing postponements of elective pediatric cardiac procedures: analysis and implementation of a discrete event simulation model. The Annals of thoracic surgery, 99(4), 1386–1391.

Dehmer, S. P., Baker-Goering, M. M., Maciosek, M. V., Hong, Y., Kottke, T. E., Margolis, K. L., & Thomas, A. J. (2016). Modeled health and economic impact of team-based care for hypertension. American journal of preventive medicine, 50(5), S34-S44.

De Mauro, A., Greco, M., & Grimaldi, M. (2015, February). What is big data? A consensual definition and a review of key research topics. In G. Giannakopoulos, D. P. Sakas, & D. Kyriaki-Manessi (Eds.), AIP conference proceedings (Vol. 1644, No. 1, pp. 97–104). AIP.

DeRienzo, C. M., Shaw, R. J., Meanor, P., Lada, E., Ferranti, J., & Tanaka, D. (2016). A discrete event simulation tool to support and predict hospital and clinic staffing. Health informatics journal, 1460458216628314.

Deutsch, E. S., Dong, Y., Halamek, L. P., Rosen, M. A., Taekman, J. M., & Rice, J. (2016). Leveraging Health Care Simulation Technology for Human Factors Research: Closing the Gap Between Lab and Bedside. Human factors, 58(7), 1082–1095.

Draycott, T. J., Crofts, J. F., Ash, J. P., Wilson, L. V., Yard, E., Sibanda, T., & Whitelaw, A. (2008). Improving neonatal outcome through practical shoulder dystocia training. Obstetrics & Gynecology, 112(1), 14–20.

Duncan, D. R., Morgenthaler, T. I., Ryu, J. H., & Daniels, C. E. (2009). Reducing iatrogenic risk in thoracentesis: establishing best practice via experiential training in a zero-risk environment. CHEST Journal, 135(5), 1315–1320.

Endo, K., Sata, N., Ishiguro, Y., Miki, A., Sasanuma, H., Sakuma, Y., & Yasuda, Y. (2014). A patient-specific surgical simulator using preoperative imaging data: an interactive simulator using a three-dimensional tactile mouse. Journal of Computational Surgery, 1(1), 10.

Evans, C. H., & Schenarts, K. D. (2016). Evolving educational techniques in surgical training. Surgical Clinics of North America, 96(1), 71–88.

Feher, M., Harris-St John, K., & Lant, A. (1991). Blood pressure measurement by junior hospital doctors–a gap in medical education?. Health trends, 24(2), 59–61.

Feldman, L. S., Sherman, V., & Fried, G. M. (2004). Using simulators to assess laparoscopic competence: ready for widespread use?.

Ferraro, N. M., Reamer, C. B., Reynolds, T. A., Howell, L. J., Moldenhauer, J. S., & Day, T. E. (2015). Capacity Planning for Maternal–Fetal Medicine Using Discrete Event Simulation. American journal of perinatology, 32(08), 761–770.

Gaba, D. M. (2004). The future vision of simulation in health care. Quality and safety in Health care, 13(suppl 1), i2-i10.

Gaba, D. M., & DeAnda, A. (1988). A comprehensive anesthesia simulation environment: re-creating the operating room for research and training. Anesthesiology, 69(3), 387–394.

Gaba, D. M., Howard, S. K., Fish, K. J., Smith, B. E., & Sowb, Y. A. (2001). Simulation-based training in anesthesia crisis resource management (ACRM): a decade of experience. Simulation & Gaming, 32(2), 175–193.

Gallagher, A. G., & Cates, C. U. (2004). Virtual reality training for the operating room and cardiac catheterisation laboratory. The Lancet, 364(9444), 1538–1540.

General Medical Council. Education Committee. (1993). Tomorrow's doctors: recommenda-tions on undergraduate medical education. London: General Medical Council.

Gerolemou, L., Fidellaga, A., Rose, K., Cooper, S., Venturanza, M., Aqeel, A., & Khouli, H. (2014). Simulation-based training for nurses in sterile techniques during central vein catheterization. American Journal of Critical Care, 23(1), 40–48.

Gettman, M. T., Blute, M. L., Chow, G. K., Neururer, R., Bartsch, G., & Peschel, R. (2004). Robotic-assisted laparoscopic partial nephrectomy: technique and initial clinical experience with DaVinci robotic system. Urology, 64(5), 914–918.

Gordon, M. S. (1974). Cardiology patient simulator: development of an animated manikin to teach cardiovascular disease. The American journal of cardiology, 34(3), 350–355.

Gordon, M. S., Forker, A. D., Gessner, I., McGuire, C., Mayer, J. W., Sajid, D. P. A., & Waugh, R. A. (1980). Teaching bedside cardiologic examination skills using "Harvey", the cardiology patient simulator. Medical Clinics of North America, 64(2), 305–313.

Gorman, L., Castiglioni, A., Hernandez, C., Asmar, A., Cendan, J., & Harris, D. (2015). Using preclinical high-fidelity medical simulations to integrate pharmacology and physiology with clinical sciences. Medical Science Educator, 25(4), 521–532.

Gordon, J. A., Brown, D. F., & Armstrong, E. G. (2006). Can a simulated critical care encounter accelerate basic science learning among preclinical medical students? A pilot study. Simulation in healthcare, 1(Inaugural), 13–17.

Grenvik, A., & Schaefer, J. (2004). From Resusci-Anne to Sim-Man: the evolution of simulators in medicine. Critical care medicine, 32(2), S56-S57.

Griswold-Theodorson, S., Ponnuru, S., Dong, C., Szyld, D., Reed, T., & McGaghie, W. C. (2015). Beyond the simulation laboratory: a realist synthesis review of clinical outcomes of simulation-based mastery learning. Academic Medicine, 90(11), 1553–1560.

Harden, R. M., & Gleeson, F. A. (1979). Assessment of clinical competence using an objective structured clinical examination (OSCE). Medical education, 13(1), 39–54.

Hoot, N. R., LeBlanc, L. J., Jones, I., Levin, S. R., Zhou, C., Gadd, C. S., & Aronsky, D. (2008). Forecasting emergency department crowding: a discrete event simulation. Annals of emergency medicine, 52(2), 116–125.

Hripcsak, G., Bakken, S., Stetson, P. D., & Patel, V. L. (2003). Mining complex clinical data for patient safety research: a framework for event discovery. Journal of biomedical informatics, 36(1), 120–130.

Huang, G. C., McSparron, J. I., Balk, E. M., Richards, J. B., Smith, C. C., Whelan, J. S., & Smetana, G. W. (2016). Procedural instruction in invasive bedside procedures: a systematic review and meta-analysis of effective teaching approaches. BMJ quality & safety, 25(4), 281–294.

Huang, T., Lan, L., Fang, X., An, P., Min, J., & Wang, F. (2015). Promises and challenges of big data computing in health sciences. Big Data Research, 2(1), 2–11.

Hunt, E. A., Shilkofski, N. A., Stavroudis, T. A., & Nelson, K. L. (2007). Simulation: translation to improved team performance. Anesthesiology clinics, 25(2), 301–319.

Ilgen, J. S., Ma, I. W., Hatala, R., & Cook, D. A. (2015). A systematic review of validity evidence for checklists versus global rating scales in simulation based assessment. Medical education, 49 (2), 161–173.

Isern, D., & Moreno, A. (2016). A systematic literature review of agents applied in healthcare. Journal of medical systems, 40(2), 43.

Issenberg, SB., Mcgaghie, W. C., Petrusa, E. R., Lee Gordon, D., & Scalese, R. J. (2005). Features and uses of high-fidelity medical simulations that lead to effective learning: a BEME systematic review. Medical teacher, 27(1), 10–28.

Johnson, O. A., Hall, P. S., & Hulme, C. (2016). NETIMIS: Dynamic Simulation of Health Economics Outcomes Using Big Data. PharmacoEconomics, 34(2), 107–114.

Khanduja, V., Lawrence, J. E., & Audenaert, E. (2016). Testing the Construct Validity of a Virtual Reality Hip Arthroscopy Simulator. Arthroscopy: The Journal of Arthroscopic & Related Surgery.

Keifenheim, K. E., Teufel, M., Ip, J., Speiser, N., Leehr, E. J., Zipfel, S., & Herrmann-Werner, A. (2015). Teaching history taking to medical students: a systematic review. BMC medical education, 15(1), 159.

Kimura, T., Morita, A., Nishimura, K., Aiyama, H., Itoh, H., Fukaya, S., & Ochiai, C. (2009). Simulation of and training for cerebral aneurysm clipping with 3Dimensional models. Neurosurgery, 65(4), 719–726.

Klevens, R. M., Edwards, J. R., Richards, C. L., Horan, T. C., Gaynes, R. P., Pollock, D. A., & Cardo, D. M. (2007). Estimating health care-associated infections and deaths in US hospitals, 2002. Public health reports, 122(2), 160–166.

Kneebone, R. (2005). Evaluating clinical simulations for learning procedural skills: a theory-based approach. Academic medicine, 80(6), 549–553.

Kohn, L. T., Corrigan, J. M., & Donaldson, M. S. (2002). To err is human: building a safer health system. National Academy of Science, Institute of Medicine, 6.

Kolominsky-Rabas, P. L., Kriza, C., Djanatliev, A., Meier, F., Uffenorde, S., Radeleff, J., & Adamson, P. B. (2016). Health economic impact of a pulmonary artery pressure sensor for heart failure telemonitoring: a dynamic simulation. Telemedicine and e-Health, 22(10), 798–808.

Konge, L., Arendrup, H., von Buchwald, C., & Ringsted, C. (2011). Virtual reality simulation of basic pulmonary procedures. Journal of bronchology & interventional pulmonology, 18(1), 38–41.

Kotsis, S. V., & Chung, K. C. (2013). Application of See One, Do One, Teach One Concept in Surgical Training. Plastic and reconstructive surgery, 131(5), 1194.

Larsen, C. R., Oestergaard, J., Ottesen, B. S., & Soerensen, J. L. (2012). The efficacy of virtual reality simulation training in laparoscopy: a systematic review of randomized trials. Acta obstetricia et gynecologica Scandinavica, 91(9), 1015–1028.

Lind, B. (2007). The birth of the resuscitation mannequin, Resusci Anne, and the teaching of mouth to mouth ventilation. Acta Anaesthesiologica Scandinavica, 51(8), 1051–1053.

Makiyama, K., Nagasaka, M., Inuiya, T., Takanami, K., Ogata, M., & Kubota, Y. (2012). Development of a patient specific simulator for laparoscopic renal surgery. International Journal of Urology, 19(9), 829–835.

Marshall, D. A., Burgos-Liz, L., Pasupathy, K. S., Padula, W. V., IJzerman, M. J., Wong, P. K., & Osgood, N. D. (2016). Transforming healthcare delivery: Integrating dynamic simulation modelling and big data in health economics and outcomes research. PharmacoEconomics, 34 (2), 115–126.

McGaghie, W. C., Issenberg, S. B., Petrusa, E. R., & Scalese, R. J. (2010). A critical review of simulation based medical education research: 2003–2009. Medical education, 44(1), 50–63.

McGaghie, W. C., Draycott, T. J., Dunn, W. F., Lopez, C. M., & Stefanidis, D. (2011a). Evaluating the impact of simulation on translational patient outcomes. Simulation in healthcare: journal of the Society for Simulation in Healthcare, 6(Suppl), S42.

McGaghie, W. C., Issenberg, S. B., Cohen, M. E. R., Barsuk, J. H., & Wayne, D. B. (2011b). Does simulation-based medical education with deliberate practice yield better results than traditional clinical education? A meta-analytic comparative review of the evidence. Academic medicine: journal of the Association of American Medical Colleges, 86(6), 706.

McGinnis, J. M., Stuckhardt, L., Saunders, R., & Smith, M. (Eds.). (2013). Best care at lower cost: the path to continuously learning health care in America. National Academies Press.

Michelson, J. D., & Manning, L. (2008). Competency assessment in simulation-based procedural education. The American Journal of Surgery, 196(4), 609–615.

Moorthy, K., Vincent, C., & Darzi, A. (2005). Simulation based training.

Morey, J. C., Simon, R., Jay, G. D., Wears, R. L., Salisbury, M., Dukes, K. A., & Berns, S. D. (2002). Error reduction and performance improvement in the emergency department through formal teamwork training: evaluation results of the MedTeams project. Health services research, 37(6), 1553–1581.

Nestel, D., Groom, J., Eikeland-Husebø, S., & O'donnell, J. M. (2011). Simulation for learning and teaching procedural skills: the state of the science. Simulation in Healthcare, 6(7), S10-S13.

Newble, D. (2004). Techniques for measuring clinical competence: objective structured clinical examinations. Medical education, 38(2), 199–203.

Opoku-Anane, J., Misa, N. Y., Li, M., Vargas, M. V., Chidimma, E., Robinson, J. K., & Moawad, G. (2015). Simulation Based Robotics Training to Test Skill Acquisition and Retention. Journal of Minimally Invasive Gynecology, 22(6), S33.

Owen, H. (2012). Early use of simulation in medical education. Simulation in Healthcare, 7(2), 102–116.

Park, J., MacRae, H., Musselman, L. J., Rossos, P., Hamstra, S. J., Wolman, S., & Reznick, R. K. (2007). Randomized controlled trial of virtual reality simulator training: transfer to live patients. The American journal of surgery, 194(2), 205–211.

Passiment, M., Sacks, H., & Huang, G. (2011). Medical simulation in medical education: results of an AAMC survey. Washington, DC: Association of American Medical Colleges.

Patrício, M. F., Julião, M., Fareleira, F., & Carneiro, A. V. (2013). Is the OSCE a feasible tool to assess competencies in undergraduate medical education?. Medical teacher, 35(6), 503–514.

Pennathur, P. R., Cao, D., Sui, Z., Lin, L., Bisantz, A. M., Fairbanks, R. J., & Wears, R. L. (2010). Development of a simulation environment to study emergency department information technology. Simulation in healthcare: journal of the Society for Simulation in Healthcare, 5(2), 103.

Phipps, M. G., Lindquist, D. G., McConaughey, E., O'brien, J. A., Raker, C. A., & Paglia, M. J. (2012). Outcomes from a labor and delivery team training program with simulation component. American journal of obstetrics and gynecology, 206(1), 3–9.

Riley, W., Davis, S., Miller, K., Hansen, H., Sainfort, F., & Sweet, R. (2011). Didactic and simulation nontechnical skills team training to improve perinatal patient outcomes in a community hospital. The Joint Commission Journal on Quality and Patient Safety, 37(8), 357–364.

Riojas, M., Feng, C., Hamilton, A., & Rozenblit, J. (2011). Knowledge elicitation for performance assessment in a computerized surgical training system. Applied Soft Computing, 11(4), 3697–3708.

Rosen, K. R., McBride, J. M., & Drake, R. L. (2009). The use of simulation in medical education to enhance students' understanding of basic sciences. Medical Teacher, 31(9), 842–846.

Rodriguez-Paz, J., Kennedy, M., Salas, E., Wu, A. W., Sexton, J. B., Hunt, E. A., & Pronovost, P. J. (2009). Beyond "see one, do one, teach one": toward a different training paradigm. Quality and Safety in Health Care, 18(1), 63–68.

Rogers, G. M., Oetting, T. A., Lee, A. G., Grignon, C., Greenlee, E., Johnson, A. T., & Carter, K. (2009). Impact of a structured surgical curriculum on ophthalmic resident cataract surgery complication rates. Journal of Cataract & Refractive Surgery, 35(11), 1956–1960.

Rotter, T., Kinsman, L., James, E., Machotta, A., Willis, J., Snow, P., & Kugler, J. (2012). The effects of clinical pathways on professional practice, patient outcomes, length of stay, and hospital costs: Cochrane systematic review and meta-analysis. Evaluation & the health professions, 35(1), 3–27.

Rozenblit, J., & Sametinger, J. (2015, April). Models in healthcare simulation: typology and security issues. In Proceedings of the Symposium on Modeling and Simulation in Medicine (pp. 31–35). Society for Computer Simulation International.

Rutherford-Hemming, T. (2012). Simulation methodology in nursing education and adult learning theory. Adult Learning, 23(3), 129–137.

Ryan, J. R., Almefty, K. K., Nakaji, P., & Frakes, D. H. (2016). Cerebral aneurysm clipping surgery simulation using patient-specific 3D printing and silicone casting. World neurosurgery, 88, 175–181.

Sametinger, J., & Rozenblit, J. (2016). Security Scores for Medical Devices. In Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016)-Volume (Vol. 5, pp. 533–541).

Sant'Ana, G. M., Cavalini, W., Negrello, B., Bonin, E. A., Dimbarre, D., Claus, C., & Salvalaggio, P. R. (2016). Retention of laparoscopic skills in naive medical students who underwent short training. Surgical Endoscopy, 1–8.

Sawyer, T., White, M., Zaveri, P., Chang, T., Ades, A., French, H., & Kessler, D. (2015). Learn, see, practice, prove, do, maintain: an evidence-based pedagogical framework for procedural skill training in medicine. Academic Medicine, 90(8), 1025–1033.

Schijven, M. P., Jakimowicz, J. J., Broeders, I. A. M. J., & Tseng, L. N. L. (2005). The Eindhoven laparoscopic cholecystectomy training course—improving operating room performance using virtual reality training: results from the first EAES accredited virtual reality trainings curriculum. Surgical Endoscopy and other Interventional Techniques, 19(9), 1220–1226.

Schout, B. M. A., Hendrikx, A. J. M., Scheele, F., Bemelmans, B. L. H., & Scherpbier, A. J. J. A. (2010). Validation and implementation of surgical simulators: a critical review of present, past, and future. Surgical endoscopy, 24(3), 536–546.

Schwid, H. A., Rooke, G. A., Michalowski, P., & Ross, B. K. (2001). Screen-based anesthesia simulation with debriefing improves performance in a mannequin-based anesthesia simulator. Teaching and learning in medicine, 13(2), 92–96.

Scott, D. J. (2006). Patient safety, competency, and the future of surgical simulation. Simulation in Healthcare, 1(3), 164–170.

Seymour, N. E. (2008). VR to OR: a review of the evidence that virtual reality simulation improves operating room performance. World journal of surgery, 32(2), 182–188.

Sedlack, R. E., & Kolars, J. C. (2004). Computer simulator training enhances the competency of gastroenterology fellows at colonoscopy: results of a pilot study. The American journal of gastroenterology, 99(1), 33–37.

Shewokis, P. A., Shariff, F. U., Liu, Y., Ayaz, H., Castellanos, A., & Lind, D. S. (2016). Acquisition, retention and transfer of simulated laparoscopic tasks using fNIR and a contextual interference paradigm. The American Journal of Surgery.

Sinha, N., Dauwels, J., Kaiser, M., Cash, S. S., Westover, M. B., Wang, Y., & Taylor, P. N. (2016). Predicting neurosurgical outcomes in focal epilepsy patients using computational modelling. Brain, aww299.

Steinemann, S., Berg, B., Skinner, A., DiTulio, A., Anzelon, K., Terada, K., & Speck, C. (2011). In situ, multidisciplinary, simulation-based teamwork training improves early trauma care. Journal of surgical education, 68(6), 472–477.

Vakharia, V. N., Vakharia, N. N., & Hill, C. S. (2016). Review of 3-dimensional printing on cranial neurosurgery simulation training. World neurosurgery, 88, 188–198.

Zafari, Z., Bryan, S., Sin, D. D., Conte, T., Khakban, R., & Sadatsafavi, M. (2016). A Systematic Review of Health Economics Simulation Models of Chronic Obstructive Pulmonary Disease. Value in Health.

Zeigler, B. P. (1976). Theory of modelling and simulation.

Zendejas, B., Brydges, R., Wang, A. T., & Cook, D. A. (2013). Patient outcomes in simulation-based medical education: a systematic review. Journal of general internal medicine, 28(8), 1078–1089.

Zendejas, B., Ruparel, R. K., & Cook, D. A. (2016). Validity evidence for the Fundamentals of Laparoscopic Surgery (FLS) program as an assessment tool: a systematic review. Surgical endoscopy, 30(2), 512–520.

Zhu, Z., Hoon Hen, B., & Liang Teow, K. (2012). Estimating ICU bed capacity using discrete event simulation. International Journal of health care quality assurance, 25(2), 134–144.

Ziv, A., Wolpe, P. R., Small, S. D., & Glick, S. (2003). Simulation based medical education: an ethical imperative. Academic Medicine, 78(8), 783–788.

Ziv, A., Ben-David, S., & Ziv, M. (2005). Simulation based medical education: an opportunity to learn from errors. Medical teacher, 27(3), 193–199.

Author Biographies

Hannes Prescher is a Research Specialist at the Arizona Simulation Technology and Education Center (ASTEC) at the University of Arizona, College of Medicine. His research interests include medical error prevention, medical, and surgical device development and testing and the translation of medical simulation into clinical outcomes. He received his BS at the University of California and completed his MD at the University of Arizona, College of Medicine. He is now a first-year resident in the Section of Plastic and Reconstructive Surgery at the University of Chicago.

Allan J. Hamilton is a Professor in the Department of Surgery at the University of Arizona. He also holds appointments in Radiation Oncology, Psychology, and Electrical and Computer Engineering. In 1995, Dr. Hamilton would become the *first American surgeon to ever receive the prestigious Lars Leksell Award for* "*pioneering discoveries*" *in the field of neurosurgery*. Since its inception, more than 250,000 patients have now been treated with extracranial SRS and it is now routinely offered at more than 100 sites in the US alone. Dr. Hamilton serves as Executive Director of the Arizona Simulation Technology and Education Center, a multidisciplinary think-tank at the Arizona Health Sciences Center devoted to developing new technologies and training procedures to reduce preventable medical adverse events. The facility serves all four healthcare-related colleges, namely, Medicine, Nursing, Public Health, and Pharmacy and provides training for 10 residencies and 3 fellowship programs. The center conducts research funded by the NIH, the NSF, as well as the Department of Defense and carries out numerous quality assurance projects in conjunction with Banner Health.

Jerzy W. Rozenblit is University Distinguished Professor, Raymond J. Oglethorpe Endowed Chair in the Electrical and Computer Engineering (ECE) Department, with as joint appointment as Professor of Surgery in the College of Medicine at The University of Arizona. From 2003 to 2011 he served as the ECE Department Head. During his tenure at the University of Arizona, he established the Model-Based Design Laboratory with major projects in design and analysis of complex, computer-based systems, hardware/software codesign, and simulation modeling. The projects have been funded by the National Science Foundation, US Army, Siemens, Infineon Technologies, Rockwell, McDonnell Douglas, NASA, Raytheon, and Semiconductor Research Corporation. Dr. Rozenblit has been active in professional service in capacities ranging from editorship of ACM, IEEE, and Society for Computer Simulation Transactions, program and general chairmanship of major conferences, to participation in various university and departmental committees. He had served as a research scientist and visiting professor at Siemens AG and Infineon AG Central Research and Development Laboratories in Munich, where over he was instrumental in the development of design frameworks for complex, computer-based systems. Currently, jointly with the Arizona Surgical Technology and Education Center, he is developing computer-guided training methods and systems for minimally invasive surgery. Co-author of several edited monographs and over two hundred publications, Jerzy holds the Ph.D. and M.Sc. degrees in Computer Science from Wayne State University, Michigan. He presently serves as Director of the *Life-Critical Computing Systems.* Initiative, a research enterprise intended to improve the reliability and safety of technology in healthcare and life-critical applications.

Part IV
Social Sciences and Management

Chapter 11
Flipping Coins and Coding Turtles

The Evolution of M&S in the Social Sciences

David C. Earnest and Erika Frydenlund

Abstract Nearly four decades ago, Thomas Schelling used coins and a checkerboard to simulate how simple social rules could produce stark neighborhood segregation. That early social science model marked the beginning of a movement to incorporate simulation into social science that continues to gain momentum today. Using political science and international studies as a frame of reference, this chapter explores the incomplete permeation of simulation into the statistical and qualitative research toolkits of those pursuing social inquiry. We begin by chronicling the development of several key advancements in modeling social systems, including formal modeling such as game theory, the adoption of statistical and computer-based modeling, and the advancement of computational social sciences using evolutionary computation and other dynamic modeling paradigms. Then, we discuss how and why simulation remains at the periphery of social science research methodologies. We compare a classic Prisoner's Dilemma model to one designed using an agent-based simulation approach to illustrate the population ecology of emergent strategies. The chapter concludes with a discussion on the ways simulation of social systems would have to evolve to have more impact on the field of social sciences.

Keywords Computational social science · Social simulation · History · Emergence · Empiricism · Behavioralism · Game theory · Computer-based simulation · System dynamics · Complex systems theory · Complexity · Sugarscape · Prisoner's dilemma · Social structures · Agent memory · Social science · Validation · Verification · Participatory methods · Model docking

D.C. Earnest
Department of Political Science & Geography, Old Dominion University, Norfolk, USA

E. Frydenlund (✉)
Virginia Modeling, Analysis and Simulation Center, Old Dominion University,
1030 University Boulevard, Suffolk, VA 23435, USA
e-mail: efrydenl@odu.edu

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_11

11.1 Introduction

In his 1979 book *Micromotives and Macrobehavior*, social scientist Thomas Schelling introduces a rudimentary but important model of residential segregation. Using a checkerboard whose squares he randomly filled with pennies and dimes, Schelling (1978) moved each coin according to a simple rule: a penny or quarter would move to an available adjacent square if it was “unhappy”—that is, if the coin was a minority among the coins on the neighboring eight squares. Of course, as each coin moved to a new square, it may change the calculus of its neighbor’s happiness, illustrating the interdependent choices of all the coins on the board. Nearly forty years old now, Schelling’s pennies and dimes represent an early example of the synergy between game theory and agent-based modeling, albeit in a manual rather than computational form. By varying the threshold of a coin’s unhappiness (for example, a coin is unhappy if fewer than 25% of adjacent squares are filled with like coins), Schelling’s model produced an astonishing result. Even in a “society” of coins that would tolerate being in the minority in their own neighborhood, stark patterns of segregation emerged after a few rounds of movement of the coins. Now considered a classic, Shelling’s segregation model illustrated an important point of social theory: individual rational choices (seeking like neighbors) may produce a collective outcome that no one intends (segregation). More generally, the model illustrates the perils of the ecological inference problem: attributing motives to individuals based on observation of macro-social dynamics. Beyond these important points of social theory, however, Schelling’s simple model also can help us think about how quantitative and simulation methodologies have contributed to the social sciences.

In this chapter, we examine the role of modeling and simulation methodologies in the social sciences. Starting with a brief history of the social sciences, we trace the origins of contemporary methods to the emergence of the study of societies as a scientific endeavor in the nineteenth century to the twentieth century innovations in social statistics, game theory, and computational methods. This history documents the considerable contributions of formal, empirical and mathematical modeling to social theory in a variety of disciplines. To illustrate the emerging convergence of various simulation methodologies including game theory, agent-based modeling (ABM) and evolutionary computation, we provide a simulation of a classic social choice problem: the prisoner’s dilemma. Our model illustrates how simulation may elaborate traditional games of social choice by extending the game to multiple players; by examining the roles of iteration and learning; and by experimentally varying the social structure in which players interact. Although merely illustrative, we argue the ABM demonstrates how modern computational methods help social scientists produce models of social phenomena that are richer, more generalizable, and more tractable than previous formal and statistical methods of social analysis. We conclude the chapter with some informed speculation about the future of modeling and simulation in the social sciences.

11.2 History of Quantitative Methods in the Social Sciences

Since their emergence as scientific fields in the 19th century, the social sciences have adopted various methods of quantitative analysis varying from systematic empirical inquiry in the nineteenth century to today's computational social simulation. As in other scientific fields, the social sciences often have benefitted from simultaneous innovations in complementary methodological fields, first statistics and then more recently low-cost computation. In this section, we provide a brief overview of this history by focusing on three broad paradigms of quantitative social scientific research: empiricism and behavioralism; mathematics and game theory; and computer-based simulation.

11.2.1 Empiricism and Behavioralism

The social sciences first emerged in the 19th century when founders of the field of sociology, principally French scholar Émile Durkheim, proposed that observers of social life should bring to their subjects the methods of science (Durkheim and Lukes 1982). Durkheim advocated for an inductive approach to building social theory based on systematic observation and inquiry. The founders of social science argued that regularities exist in social life that researchers can discover, observe and measure. As in the natural and physical sciences, early social scientists espoused a rigor of methodology that would permit not only the verification or falsification of social theories but also the replication of results. Their emphasis on non-normative social sciences—theorization, observation and inference free from the value judgments of scholars—represented an important disjuncture from social theory's origins in the humanities. Although social philosophy and history remain important (normative) fields of inquiry in modern social scientific disciplines, the predominance of positivist epistemologies in sociology, anthropology, political science, economics and psychology attests to the enduring influence of the scientific method on social research.

The earliest quantitative methods in the social sciences were large-sample surveys, the data from which scholars would create and/or validate social theories. Durkheim's *Suicide* (1897) is an early example of this inductive, data-driven approach. Using data on suicides from different police districts, Durkheim hypothesized that stronger social "control" (norms and values) among Catholic communities explained their lower suicide rates when compared to Protestant communities. Although Durkheim's study subsequently faced considerable criticism, it typifies early quantitative methods in social science: it used systematically collected data to argue that properties of societies (social control) rather than individual or psychological factors can explain observed differences in societies. In this respect, early survey research proposed a social structural ontology that one can observe and measure; that exists external to the individual; and that is fundamentally different than the simple aggregation of individual preferences or choices.

In the mid-twentieth century, however, some social scientists began to question the structural theories of social research. Originating in political science, behavioralism accepted the positivist methods of social research including verifiability, systematic measurement, replicability and non-normative theory. However, behavioralists proposed that social inquiry should re-focus on individuals rather than social practices and institutions. By examining how people process information, make decisions, and communicate with each other, behavioralism distinguished itself from the structural emphases that characterized the social sciences in the first half of the twentieth century. In addition to the discipline of political science, psychology also adopted behavioral methods. More generally, the behavioral sciences differ from the social sciences in their greater emphasis on observation of, measurement of, and theorization based on individuals rather than the properties of social groups. Although behavioralism and social structuralism differed in ontologies, theories, and hypotheses, both benefitted from innovations that marked the emergence of statistics as a discipline distinct from mathematics. In particular, sociologist Paul Lazarsfeld, who founded Columbia University's Bureau of Applied Social Research, pioneered the statistical analysis of survey data and latent class analysis (Clogg 1992). In general, during the mid-twentieth century social scientists increasingly combined the systematized empirical methods of early social research with the inferential methods of statistics (Blalock 1974).

11.2.2 Game Theory

Just as behavioralism was a reaction to the structuralism of early social research, so was game theory a reaction to the inductive empiricism of most social sciences. Game theory is a set of formal (mathematical) methods for understanding bargaining, conflict, and cooperation among rational actors who have interdependent "payoffs" or rewards for their choices. Rather than constructing social theory through induction based on empirical observation of regularities in social actors, game theory proposes to deduce social behaviors from formal representations of actor choices, incentives, and rewards. Game theorists propose that social choices involve uncertain outcomes ("gambles"); interdependent rewards (payoffs); and variable amounts of uncertainty about these choices. By representing gambles, payoffs, and information in mathematical equations, game theory proposes that social scientists can deduce likely choices of individuals and, by extension, the prospects for social cooperation or conflict.

Most historians of the social sciences date the origins of modern game theory to discussions in the 1940s between mathematicians Oskar Morgenstern and John von Neumann, both of whom worked at Princeton University's Institute for Advanced Studies. O'Rand (1992) argues that the historical context helps understand both the motivation for a deductive social science, and its evolution from an obscure branch to a widely practiced methodology in the social sciences. During and following World War II, federally sponsored research in the United States emphasized problems of war and peace as well as industrialization and scientific management (which found expression as operations research). In addition, tumult in Europe led to the migration of a considerable number of scientists to the United States, including Morgenstern (an Austrian who was visiting Princeton when Nazi Germany invaded Austria) and von Neumann (a Hungarian who left a position in Germany for Princeton in 1930). In 1944, von Neumann and Morgenstern published *The Theory of Games and Economic Behavior* which introduced the concept of expected utility, or the subjective valuation that a rational actor attributes to a choice characterized by uncertainty (Von Neumann and Morgenstern 1944). Von Neumann–Morgenstern utility theory remains the foundation of contemporary game theory. Although scientists at the Institute for Advanced Study engaged in productive exchange and collaboration, O'Rand argues that the relatively insular community (and another at the University of Michigan) shared ideas and innovations through informal and social communications more than through scholarly publishing (O'Rand 1992). For this reason, adoption of game theory in the social sciences proceeded relatively slowly, only finding a broader audience in the late 1950s.

Among the challenges of early game theory was how to model players' knowledge about a game's structure of payoffs and its history of play. Although early models of games assumed that players would make simultaneous choices, the innovation of sequential play games required modelers to explicate whether players know the history of prior plays (what they call "perfect" information) or the strategies and payoffs available to other players ("complete" information) (Gibbons 1992). Another Hungarian-born mathematician, John Harsanyi, made substantial contributions to the study of games of incomplete information (Harsanyi 1967). Together with von Neumann and Morgenstern, Harsanyi's contributions form the foundation of modern game theory. Their deductive methods allow researchers to model "static" games of simultaneous play; dynamic games of sequential play; with perfect, imperfect, complete and incomplete information.

Whereas in the 1950s game theory remained the province of mathematically inclined social scientists, two more recent works contributed to the broad adoption of the methodology in all social scientific disciplines. The first was Thomas Schelling's *The Strategy of Conflict* (1960), which relaxed the assumption of von Neumann–Morgenstern utility theory to hypothesize that irrational actors and credible threats to cheat could alter equilibrium solutions to games. In this respect, Schelling brought to game theorists important discussions about player motives including fear, honor, and myopic perception. The other important contribution arises from a series of studies conducted by political scientist Robert Axelrod on the prisoner's dilemma, one of the canonical static games of complete information in which players face strong incentives to eschew cooperation ("defect" in the argot of game theory). Axelrod's (1980) research, including a tournament in which he invited fellow scientists to propose optimal strategies for the prisoner's dilemma, culminated in the publication of *The Evolution of Cooperation* (1984), which illustrated that repeated plays of the prisoner's dilemma allows players to learn cooperative strategies that improve their long-term payoffs. This finding has informed not only a vast subsequent literature on how social institutions produce cooperation but has also informed policy-making in diverse fields.

O'Rand notes that while at the Institute for Advanced Study, Morgenstern was "a pariah among the traditional economists on the faculty at Princeton" (O'Rand 1992: 184–85). In the half decade since its wide adoption in the social sciences, several game theorists have received the Nobel Prize in Economics including Harsanyi, John Nash and Reinhard Selten (in 1994); Robert Lucas (in 1995); and Schelling (in 2005). The recognition of these scholars attests to the widespread influence of game theory in the social sciences today. Along with statistics, it remains a foundational methodology in most graduate curricula in the social sciences.

11.3 Computer-Based Simulation

In the 1990s, computer-based simulation for social science research questions began to grow in popularity as a research method. The 1990s ended with the establishment of the Journal of Artificial Societies and Social Simulation (JASSS) in 1998. This move paved the way for broader acceptance of M&S in the social sciences by serving as a platform for interdisciplinary research. Two of the main modeling paradigms, system dynamics and agent-based modeling, are discussed below with primary emphasis on ABM.

11.3.1 System Dynamics

Economists paved the way for system dynamics modeling in the social sciences since the 1970s, with computational models giving way to system dynamics (SD) models of global socio-political and economic interactions (Chadwick 2000). The International Futures model, begun in the 1980s, links data on countries grouped by region and evaluates factors such as economies, demographics, and food and energy systems for policy analysis (Hughes 1999). In that era of advancing computer-based modeling technologies, anthropological research utilizing system dynamics approaches also appeared (Picardi 1976), but with less influence than the economic models. Early on, some even proposed that SD models could serve as learning tools for articulating processes informed by qualitative data in the social sciences (Wolstenholme and Coyle 1983) and the study of history (Bianchi 1989). The purposes of these models, as of the late 1990s, was largely to plan or conceptualize complex processes rather than to predict or test hypotheses (Chadwick 2000).

Starting in the early 2000s, more social phenomena become the subject of SD models. van Dijkum et al. (2002) introduced a system dynamics model to look at individual learning and fatigue based on survey data for health psychology research. They contend that since social scientists are familiar with evaluating phenomena using cause and effects models, and system dynamics models provide a natural way to translate these traditional research approaches into mathematical models that can accommodate both quantitative and qualitative data. Continuing with the movement toward modeling socio-economic phenomena, more contemporary models include SD approaches to understanding diffusion of democracy (Sandberg 2011), land use among agrarian societies (Rasmussen et al. 2012), housing markets (Ouml et al. 2014), and refugee migration (Vernon-Bido et al. 2017). System dynamics, while still not widely used by many social scientists, is a fruitful area for modeling complex processes.

11.3.2 Agent-Based Modeling

Agent-based modeling (ABM), however, is particularly well-suited to the field of social sciences and has experience wider acceptance than system dynamics paradigms. The accessibility of object-oriented programming languages and user-friendly environments has created a community of researchers—students and faculty alike—who embrace ABM as a method for exploring the physical and social sciences (Lerner et al. 2010). From Schelling's (1978) model of housing segregation based originally on a physical checkboard model and later in a computer algorithm, the vast variety of agent-based models is reflected in modern repositories like OpenABM (https://www.openabm.org) and the NetLogo Models Library and Modeling Commons (http://modelingcommonns.org).

Axelrod (1986) led the forefront of the social science ABM movement when he simulated the emergence of behavioral norms and firmly grounding ABM in the fields of sociology and political science. In the 1990s, Epstein and Axtell (1996) developed the *Sugarscape* model where simple behavior rules about agents consuming resources resulted in emergence of behaviors representing those that evolve in society. There were even models constructed to inform policy and practice in real-world settings in the early years of social science ABM (Doran 2001). Social scientists began steadily to contribute to the growing field of ABM as they were drawn to the ability to construct heterogeneous, autonomous agents in boundedly rational space, rather than reconstruct social realities based on traditional mathematical or statistical models (Gilbert and Troitzsch 2005; Epstein 2006; Gilbert 2008; Macy and Willer 2002).

Throughout the decades that followed this initial work, social scientists constructed agent-based models to explore and test theories on identity and norm creation (Lustick 2000; Rousseau and van der Veen 2005), ethnic violence and conflict (Bhavnani and Backer 2000; Miodownik and Cartrite 2010; Yamamoto

2015), and political processes (Lustick 2011; Epstein 2013; Bhavnani 2003; Castro 1999), among countless other social phenomena. Agar (1997a) went on to propose that ABM presents an opportunity to move beyond the dichotomy of deductive versus inductive approaches. That argument continues on in the development of ABM for educational purposes as a way to more rigorously analyze physical and social phenomena (Jacobson and Wilensky 2006; Wilensky and Resnick 1999).

11.4 Complex Systems Theory

Various modeling paradigms have begun to infiltrate the traditional statistical and qualitative methods of social sciences, but Complex Systems Theory unified many of these pursuits across disciplines. Weaver (1948) proposed that mathematical and statistical advances, while important, were not yet powerful enough to capture the complexity of many physical and social systems. He proposed a move toward exploring what cannot be quantified by traditional scientific methods, thus opening the door for complex systems thinking. In a modeling context, Wolfram (1985) used cellular automata models to demonstrate regularities that underlie complexity. He began using computer simulations and experiments to test the boundaries of complexity, try to understand where simple rules result in complex outcomes, and witness the edges of chaos. Scholars adapted these ideas to other fields, including international relations where the concepts are applied to the complex adaptive systems of political processes (Cederman 1997; Bousquet and Geyer 2011; Earnest 2015).

Relative to computational social sciences, complex systems have certain characteristics in common: simple components/agents, nonlinear interactions, emergent behavior (often through learning and evolution), and no centralized control entity (Mitchell 2009). These attributes align well with the defining features of agent-based models, specifically heterogeneity, repeated and localized interactions of autonomous agents, and bounded rationality where agents have limited global knowledge about the system in which they exist (Epstein 2006). These fundamental attributes of ABM lend themselves to generating complex social phenomena from the bottom up by endowing agents with attributes and algorithms for decision-making and learning, and then observing the macroscopic-level phenomena that arise from repeated interactions. Epstein (2006) summarizes, "The agent-based approach invites the interpretation of society as a distributed computational device, and in turn the interpretation of social dynamics as a type of computation". ABM, then, opened up a new avenue for social science research that has yet to fully reach the mainstream.

Two distinctive schools of agent-based modeling emerged in the 1990s, which one might call the North American and European traditions. Reflecting the approach of Epstein and Axtell's (1996) seminal "Sugarscape" model, the North American school views agent-based models as abstract and general representations of a complex system. There is no real-world referent for the Sugarscape, for example. Much as game theorists rely on deductive methods to model interaction, agent-based modelers in this tradition deduce agent rules and interactions from extant theory. Only after model construction and experimentation would the modeler seek to compare simulation results with empirical data. One advantage of the North American school is that modelers can investigate social phenomena for which empirical data is costly, scarce, difficult to obtain for ethical reasons, or from historically rare events. For example, researchers have used ABMs to study dynamics of secessionist movements (Lustick et al. 2004) or insurgencies (Bennett 2008). The North American school of ABM reflects the earliest abstract agent-based modeling developed at the Santa Fe Institute; Epstein and Axtell's initial collaboration and then Axtell's work at George Mason University; and a cluster of researchers at the University of Michigan including Agar's work on *The Evolution of Cooperation* (1997b), John Holland's pioneering work on genetic algorithms, and political scientist Scott Page.

By contrast, a European tradition emerged around this same time primarily in Paris and Manchester, UK. In contrast to the North American tradition, the European school embraced "evidence-driven" modeling. As the name suggests, the European school views empirical measurement and data-gathering as antecedent to model construction. Rather than an ABM serving as an abstract generalization of a social phenomenon, modelers in the European tradition view ABMs as representations of real-world social systems and problems. Modelers in this tradition often engage with stakeholders or subjects to understand their perceptions, decision-making, and strategies when confronted with a social choice. The companion modeling approach developed by Barreteau et al. (2003a) exemplifies this engagement with human subjects: the approach not only consults with stakeholders but involves them in an iterative process of model construction, feedback, and validation. Variously referred to as evidence-driven modeling or participatory modeling, this approach's emphasis on fidelity to real-world social systems provides modelers with ABMs that enjoy strong micro- and macro-level validity (i.e., at the level of both agents and emergent phenomena). Typical of this school's approach are studies that have used evidence-driven ABMs to study the management of common pool resources (e.g., Le Page and Bommel 2005). Whereas the North American tradition sacrifices some level of validity for generality, the European school prefers validity to generalization. Examples of the European school include the work of Bousquet (2005), Geller and Scott Moss (2008), and Barreteau (2003b, 2007).

Although the European and North American schools of modeling originated with distinctive approaches, today modelers productively borrow from both traditions such that the distinction is blurred. Neither tradition is a substitute for the other; they are complementary methods, the choice of which depends upon the modeler's need for validity, generalization, and empirical data.

11.5 Prisoner's Dilemma Example

Among the most studied problems of collective action, the prisoner's dilemma offers a useful metaphor for understanding challenges to social cooperation when utility-maximizing actors cannot enforce agreements and are concerned about the distribution of gains (Table 11.1). Prisoner's Dilemma is a game theoretic model involving two players attempting to cooperate to receive a lighter sentence. The premise is that two people are arrested for a crime, but the scant evidence requires a confession. Each player cannot know what the other will do. The sentencing works as follows:

- If both players choose to stay silent, they will both receive a reduced sentence, say one year each.
- If both players implicate the other, they will each receive two years in jail.
- If one prisoner implicates the other, but the other remains silent, the accuser gets off scott free while the other gets three years.

Cooperation would yield the best collective result, when both stay silent (for an aggregate payoff of two years). However, since the players cannot communicate about their choices, they are motivated to implicate each other and thus each receive two years sentencing (for an aggregate payoff of four years; see Table 11.1 for payoff matrix).

In its original two-player, single-play form—what game theorists call a static game—its very parsimony limits its application to a broad range of real-world cooperation problems. The two-player game tells us little about how large groups may cooperate, whether multinational firms seeking to set technical standards or civil society groups seeking to mobilize supporters in support of a common cause.

Early studies of the prisoner's dilemma, particularly Schelling's (1980) seminal *The Strategy of Conflict*, noted the imperative of extending the game to a multi-player and iterated form. The challenge of multiplayer games is that, once extended beyond the simple two-player structure, the modeler must make theoretically informed assumptions about the structure of social relations among the many players. In this context, social "structure" refers to the organization of relations among actors, including direct or face-to-face interactions (a spatial assumption) or indirect interactions such as among firms competing in a market (a network assumption). Such assumptions must include how actors communicate preferences among each other, and whether they play simultaneously with all other players or a series of sequential two-player games. One scholar calls this the problem of

Table 11.1 The two-player Prisoner's Dilemma with cardinal payoffs

	Prisoner's Dilemma	
Player 1	Player 2	
	Cooperate	Defect
Cooperate	2, 2	0, 3
Defect	3, 0	1, 1

"context preservation", which requires some mechanism to preserve the social neighborhood or temporal context of the players (Power 2009). Studies in multi-player prisoner's dilemma have found that outcomes of the game are highly sensitive to different assumptions about the spatial or networked structure of iterated player interactions (Skyrms and Pemantle 2009; Matsushima and Ikegami 1998).

Contemporary modelers have adopted two strategies for representing social structure in the multiplayer prisoner's dilemma. One is to use cellular automata to govern the structure of interaction among players: an actor would play the game only with its Moore neighbors (those represented spatially as above, below, left and right of the player's cell) (Nowak and May 1992; Akimov and Soutchanski 1994). The other is to use explicit representations of the physical space of game play—for example, using geographic information science (GIS) data about real-world locations of social interaction (Power 2009)—or adopting social network structures, either theoretical or empirical, to examine how network properties affect the prospects of cooperation (Earnest 2015). One advantage of the networked representation of game context is that the modeler may allow social structure to evolve endogenously in a multiplayer game. For example, players may learn to play repeatedly with "trusted" players while ignoring others, effectively rewiring the social network to find the most cooperative partners.

Because differences in social networks may profoundly affect the prospects for cooperation in the multiplayer prisoner's dilemma, we build an agent-based model that allows us to vary experimentally the network structure in which players interact. Although the formalism of social network analysis and its mathematical cousin graph theory may be daunting, the simple intuition these methods capture is that rarely do actors know everyone else in a social system, a so-called "all-to-all" network. It is much more likely for an individual to have a few close relations, who in turn have a circle of social relations that only partially overlaps with the first individual. Studies of multiplayer prisoner's dilemma may use commonly studied network structures such as a fully connected (all-to-all), small world, or scale-free network. Prior studies have found that network structure affects the prospects for cooperation. One study found that cooperation emerges if the average number of a player's neighbors (the player's degree) exceeds the ratio of the benefits to costs of action (Ohtsuki et al. 2006). Another finds that scale-free networks greatly enhance the prospects for cooperation in a variety of multiplayer games (Santos and Pacheco 2005).

Independent of a game's social structures, the number of players likely will affect the prospects for cooperation in prisoner's dilemma. Interesting, extant theory provides contradictory expectations concerning the effect of the number of players on cooperation. Classic collective action theory suggests that as the number of actors grows, problems of "free riding" (consuming the benefits of cooperation without contributing to collective efforts) overwhelm the incentives for cooperation (Olson 2009). However, some studies have found that in larger groups, players may be more likely to play Pareto-improving strategies because they are less concerned about the consequences of inequitable payoffs (Snidal 1991; Pahre 1994; Kahler 1992). Similarly, an empirical study has found that larger groups of human subjects tend to produce a more equitable and altruistic distribution of the gains from cooperation. Larger groups may make it easier for the group to detect and sanction free riders (Liebrand 1984).

Prior research suggests, then, that in multiplayer prisoner's dilemma, the prospects of cooperation depend on both the number of players of the game, and the social structure in which they interact. To test these expectations, our simulation experimentally varies both the number of game players and the social network in which they play the prisoner's dilemma game.

11.5.1 The Simulation

We constructed an agent-based model of the multiplayer prisoner's dilemma. It allows the researcher to choose the number of players, from the classic two-person game to 20 players. To explore the effect of social structure on cooperation, the model arrays players in a circle from which it constructs social networks. It incorporates four well-known social network structures: the fully connected, "all-to-all" network; a small world network (Watts and Strogatz 1998); a scale-free network (Barabási and Albert 1999); and a nearest neighbor network (i.e., the player interacts only with players to the left and right in the circle). Because prior research has found that repeated plays of the game (iteration) produce cooperative strategies (Axelrod 2006), the simulation endows each agent with a memory of previous plays of the game, which the model varies experimentally from one to five previous plays.

The game proceeds as follows: a randomly chosen agent randomly picks one partner from its social network with whom to play the PD game. Each pair plays a choice and receives a cardinal payoff, as represented in Table 11.1. Because agents randomly choose a network neighbor, the likelihood of a given agent repeatedly playing the game with the same counterparty varies inversely with the number of players. After all players have played this pairwise game once, the process repeats for a limited number of iterations that the modeler determines. Table 11.2 presents the pseudocode for the model. Strictly speaking, the simulation adopts an approach that has a series of simultaneous and parallel two-player games; a true multiplayer game would be one in which all agents play against each other simultaneously.

To model how players learn and adapted in iterated games, we implement a genetic algorithm developed by Axelrod (1987), as described by Mitchell (2009). In brief, the model encodes agent strategies as a random bit string of length 2 m^2, with m = memory of past plays. At initiation agents receive a set of strategies and play each strategy once (a "generation" of the algorithm). For each strategy, an agent plays the choice (1 = cooperate, 0 = defect) at a bit position that corresponds to the pattern of previous rounds of play. After all agents have played a round, the algorithm implements a fitness-proportionate selection routine; a crossover procedure that exchanges among selected strategies one of every four bits on average; and a mutation strategy that flips each bit with p = 0.001. Our experiment has the agents play 50 strategies per generation, for 100 generations.

Table 11.2 Pseudocode for the Multiplayer Prisoner's Dilemma

Initialization	Create N negotiator agents and distribute them in a circle Endow them with a memory of length M Seed initial memory set with random bit string of length M Endow with a network of other negotiators Network types: fully connected, nearest neighbor, small world, scale-free
Execution	Loop for 20 rounds of play: Each negotiator agent: Randomly select one neighbor to play Checks memory of game play Convert memory bit string m from binary to decimal format = history h Play choice x from position h in the strategy Record partner's choice y Receive payoff for outcome x, y for the game Add partner's choice y to the end of memory bit string End Loop
Genetic algorithm	Initialization: Endow negotiator agents with a set of 40 strategies One strategy = bit string of length 2 m^2 with $p(i = 1) = 0.5$ 40 strategies = 1 generation Loop for 40 generations: Record the mean Hamming distance of negotiators' strategies Record the percentage of cooperation plays in all negotiators' strategies Record the mean total payoffs of negotiators At the end of each generation: Select 20 strategies, using either pairwise or fitness-proportionate rule With $p = 0.25$, crossover two selected strategies at a randomly selected bit With $p = 0.001$, flip each bit of a selected strategy Add 20 strategies of length 2 m^2 with $p(i = 1) = 0.5$ Execute the game play End Loop

11.5.2 Expectations and Findings

Prior experiments with two-player prisoner's dilemma give us two expectations for the simulation. First, iteration of the game is more likely to produce cooperation. Because iteration implies players remember past plays of the game, in our implementation we expect that agents with longer memories should produce more cooperation than those with shorter memories. Second, prior studies have found that a strategy known as "tit-for-tat" is an optimal solution for the prisoner's dilemma. This strategy is a simple one: players will reciprocate the choice of their partner in the previous round. In our implementation, the ABM measures outcomes of the game as a bit string of length 2 m, with odd-numbered positions recording the agent's choice, and the even-numbered positions recording the choice of the agent's randomly chosen network partner. Given this implementation, the emergence of tit-for-tat will appear as alternating ones and zeroes in the outcome bitstring, e.g., "10101010" for a game in which agents remember the previous four rounds of play. Because there are well-known variants of tit-for-tat (e.g., tit-for-two-tats, two-tits-for-tat) we expect to find a variety of strategies with broad patterns of reciprocation of cooperation and defection.

Our experiment varied three model parameters. To test the hypothesized effect of player numbers on cooperation, we experimentally vary the number of players from two (which is the classic game) to 4, 6, 8, 10, and 20. Because we expect iteration to improve prospects for cooperation, we experimentally vary the length of players' memory from one previous round of play to 2, 3, 4 and 5 previous rounds. Finally, to examine the effect of social structure on the prospects for cooperation, we have agents play the PD game in four different social structures: fully connected; small world; scale-free; and nearest neighbor networks. The experiment thus produces 120 runs, though within each run the genetic algorithm has players learning over the course of 100 generations. We measure player strategies and outcomes only at the end of each of the 120 runs. Because we measure each player in each run, the experiment gives us 100,000 observations of players' strategies at the end of each run.

Table 11.3 reports the most frequent outcomes of the game for each value of the memory parameter. The reported results are consistent with two theoretical expectations. First, as players' memories grow longer, the frequency of the pure defection outcome (all zeroes indicate both players are choosing defect as their strategy) declines. As expected, at $m = 1$ pure defection is the most frequent outcome of the simulation, accounting for more than half of games played in the final generation. At $m = 4$, pure defection occurs in only about three percent of the games in the final generation. Pure defection is not even among the top ten most frequent outcomes when $m = 5$. These results suggest that when playing iterated games with a greater number of plays, the genetic algorithm allows players to "learn" or evolve strategies that produce more cooperation and avoid the trap of pure defection. Second, the results reported in Table 11.2 illustrate the emergence of pure tit-for-tat strategies as agents play longer iterated games. When $m = 2$, tit-for-tat ("1010" and "0101") occurs as the sixth- and seventh most frequent outcomes. Tit-for-tat is the second- and third most frequent outcomes when $m = 4$, and the first- and third most frequent outcomes when $m = 5$. Although tit-for-tat may represent a smaller percentage of the outcomes as m grows, this is because the universe of possible outcomes grows by 2 m^2. Given the very large number of possible outcomes, the relative preference of tit-for-tat as a very frequent outcome suggests that agents evolve cooperative strategies over the course of the simulation.

To examine whether the number of players and network structure affects the likelihood of cooperation, we regressed several model parameters on the average player score in the final round of each generation. Although players' scores might theoretically be higher if they play a constant defect strategy against "suckers" that play regular cooperative strategies, the genetic algorithm suggests this is unlikely as players learn not to play the sucker strategy. Because Table 11.3 found increasingly frequent tit-for-tat cooperative strategies, we can reasonably assume that higher player scores in the final generation associate with cooperative strategies. The results reported in Table 11.4 are consistent with theoretical expectations. As expected, as the number of players increases, agents tend to earn higher scores. Players' memory has a strong and positive effect, again suggesting that iteration

Table 11.3 Most frequent outcomes of the game, by memory m parameter. Tit-for-tat outcomes highlighted by outcome and strategy (Outcome) and percentage in the population (Pct.)

Memory = 1		Memory = 2		Memory = 3		Memory = 4		Memory = 5	
Outcome	Pct.	Outcome	Pct.	Outcome	Pct.	Outcome	Pct.	Outcome	Pct.
00	56.91	0000	19.86	000000	4.13	00000000	2.92	**1010101010**	**0.41**
10	18.32	0010	8.93	001000	2.81	**01010101**	**1.15**	1100000111	0.40
01	18.26	1000	8.68	000010	2.46	**10101010**	**1.04**	**0101010101**	**0.36**
11	6.51	0100	7.91	**010101**	**2.44**	00001001	0.56	1100001011	0.36
		0001	7.51	000100	2.43	10110000	0.53	0001111010	0.19
		0101	**6.90**	**101010**	**2.30**	10111100	0.52	1111010111	0.18
		1010	**6.57**	100010	2.23	00010100	0.51	0010000100	0.17
		1001	4.98	101000	2.19	00100000	0.50	0011000100	0.17
		0110	4.45	100000	2.12	10000001	0.50	0010110000	0.17
		1100	4.06	000001	2.10	10001010	0.50	10000100111	0.17

Table 11.4 Regression results, with robust standard errors

N =	100000
F(6, 99993)	12353.58
Prob > F	0
R-squared	0.4257
Adj R-squared	0.4257
Root MSE	7.3319

	Coef.	*Robust std. err.*	*t*	*P > t*	*95% conf.*	*Interval*
No. of players	0.253	0.004	56.60	<.001	0.244	0.262
Memory (m)	3.861	0.017	227.80	<.001	3.828	3.894
Small World dummy	−3.857	0.067	−57.43	<.001	−3.989	−3.725
Scale-free dummy	−3.382	0.070	−48.63	<.001	−3.518	−3.246
Nearest Neighbor dummy	−7.507	0.065	115.12	<.001	−7.635	−7.379
Generation of the GA	−0.017	0.001	−21.59	<.001	−0.019	−0.016
Constant	82.192	0.116	707.97	<.001	81.965	82.420

leads to cooperative strategies and higher scores. To assess the effect of social network structure, we treat the fully connected ("all-to-all") network as the ideal baseline: all agents play against every other agent with a uniform probability. The results in Table 11.3 indicate that different network structures have a significant effect on players' cooperation, in this case a negative effect when compared to the baseline of a fully connected social network. The nearest neighbor network is the least advantageous, with players scoring an average of seven and a half points lower in game ceteris paribus. By contrast, players in scale-free networks score significantly higher than players in the small world or nearest neighbor network structures ($t = 32.14$, $p < 0.001$).

The consistency of these results with theoretical expectations illustrates how researchers may use simulation methodologies to build upon formal models and prior empirical scholarship. Because formal models of social choice problems do not permit easy extrapolation to the multiplayers games more typical in the social realm, computer-based methodologies allow researchers to expand these formal models and to test their sensitivity to assumptions about the social structure of players. Our use of a genetic algorithm and network theory illustrates how multiple formal and computational methods may inform analyses of social problems. This integration of formal and computational methods is, in our view, an emerging paradigm that promises the theoretical breakthroughs that characterized the emergence of empirical large-sample studies in the late nineteenth century and then game theory in the mid-twentieth century.

11.6 Model Validation and Verification Challenges

Despite the power that modeling and simulation brings to the study of social sciences, major challenges remain before it will likely receive widespread acceptance as a research method. Social science, and political science in particular, relies upon humanist approaches that foreground agency of actors at the local, regional, or international levels. This is not entirely at odds with the modeling and simulation approaches presented here, but for many, it is philosophically complex to reduce social interactions and behaviors to simple algorithms. The critical realist theory-based, post-positivist views that dominate much of the social sciences see theory as revisable and the scientific method, while worthy of pursuit, as a flawed process. Much of modeling and simulation as a discipline, as well as complexity theory, seeks positivist truths and theories about the generalizable underlying causal mechanisms that shape our world. Agar (2004) offers this thought, "[Complex Adaptive System] offers an ironic combination of the poststructural and the scientific, a framework that accepts the heresy of researcher influence but then deals with it in a systematic fashion". When simulating physical systems, modeling and simulation can achieve a relatively complete picture of the system in question. For social systems, however, models are often drastically simplified and can only capture a fraction of the potential causal mechanisms driving human behavior and interactions. Though other quantitative and qualitative methods may suffer from weaknesses, the inability of M&S to capture the full range of a social context is often seen as a flaw in the methodology.

M&S also requires a large amount of empirical data to calibrate and validate models. The data-intensive requirements of most models are problematic for social sciences, depending on the type of research question (Borero and Squazzoni 2005; Janssen and Ostrom 2006). In most social sciences, sample sizes are small compared to datasets available for physical sciences. Many social data are also contextualized and temporally limited, for example, an ethnographic study over ten years of an isolated ethnic group. The data required to construct models and simulations of social systems thus are often not available. Given ABM's relative integration with social science, some scholars have advanced methods for tying these models to qualitative data. Malerba et al. (2001), for instance, propose an evolutionary economic model derived from historical analysis. Ethnographic data (Agar 2005), including through proposed Grounded Theory approaches (Neumann 2015; Dilaver 2015) have found their way into the evolution of ABM for social sciences, though not without critiques about the roles of this type of data or results (Agar 2003, 2005; Yang and Gilbert 2008). Even participatory methods common to qualitative studies have emerged as 'companion modeling' practices, where subject matter experts play a role in developing, executing, and verifying the simulation (Gurung et al. 2006; Polhill et al. 2010).

Due to the data challenges facing social science modelers, validation of these simulations also proves problematic (Ormerod and Rosewell 2009). Anthropologists have looked toward comparing models of similar phenomena to validate high-fidelity models (Kuznar 2006). This often is performed as a kind of 'model docking' where models of similar phenomena are tested to determine if they can produce similar results (Axtell et al. 1996). While not validating against real-world data, this comparison method provides some means for checking that the model is accurately reflecting the real-world phenomena as the researchers intended. To achieve a level of verification, Miller (1998) proposes using algorithms such as genetic algorithms or simulated annealing to search the parameter space of models in an attempt to "break" them or find combinations of parameters that produce unexpected results. This can serve as a method of verifying that the model is performing without overt errors (debugging), but also provide insight into particular combinations of parameters that produce emergence of macro-level effects that were unintended at the outset of the model.

These challenges are at the forefront of innovation in computational social science research. The primary goal, as with any method, is to design a study and utilize the computational tool in order to answer the research question or objectives at hand. Rather than focusing on models as predictive tools, Epstein (2008) proposes that we consider the multitude of reasons for developing models of social phenomena. Among those of particular interest to social scientists are to explain phenomena, provide a framework for systematically thinking about problems and dynamics, testing accepted theories, and grounding policy development and discussion (Epstein 2008).

11.7 Conclusions

Traditional quantitative and qualitative methods dominate the social sciences, with statistics and ethnography at the forefront. While scientists understand that these methods have limitations in terms of over-generalization by the former and ungeneralizable small sample sizes by the latter, it is difficult to break away from familiar analytical tools. As this chapter demonstrates, great strides have been made to develop methods to ground models in qualitative and limited quantitative datasets, as well as validate models in restricted data environments. Modeling and simulation, particularly ABM, holds great promise for becoming a fundamental tool in many social scientists' research toolkit as it evolves to incorporate multiple types of data and theories and programming environments become increasingly user-friendly. The advancement of the state of the art will require flexibility in the discipline of M&S as well as the social sciences. For M&S, modeling paradigms and stringent expectations for traditional forms of calibration, validation, and verification must bend to meet the non-predictive purposes of many social science models. Social scientists must also adjust their approach to research, thinking outside the bounds of traditional statistical or qualitative methods to explore how M&S can further our understanding of social phenomena.

Review Questions

1. What are some of the differences between social sciences and the natural and physical sciences? What are the implications of these differences for modeling and simulation in the social sciences?
2. Who were the important innovators in empirical social scientific methods? Game theory? Agent-based modeling?
3. What is behavioralism? How does it differ from earlier paradigms of empirical social research?
4. What is the prisoner's dilemma? What does it tell us about the relationship between individual choices and collective action?
5. What are the differences in assumptions and methods of the North American and European schools of agent-based modeling for the social sciences?
6. Methodological paradigms in the social sciences often depend upon technical innovations in scientific research. What are some examples of these technical innovations and their contributions to modeling in the social sciences?
7. What lessons might the social sciences learn from modeling and simulation in other disciplines? Conversely, what lessons might disciplines such as engineering learn from social scientists?

References

Agar, Michael. 2003. 'My kingdom for a function: Modeling misadventures of the innumerate', *Journal of Artificial Societies and Social Simulation*, 6.

Agar, Michael. 2004. 'We have met the other and we're all nonlinear: Ethnography as a nonlinear dynamic system', *Complexity*, 10: 16–24.

Agar, Michael. 2005. 'Agents in Living Color: Towards Emic Agent-Based Models', *Journal of Artificial Societies and Social Simulation*, 8.

Agar, Michael. 1984. *The evolution of cooperation* (Basic Books: New York).

Agar, Michael. 1986. 'An Evolutionary Approach to Norms', *The American Political Science Review*, 80: 1095–111.

Agar, Michael. 1987. 'The evolution of strategies in the iterated prisoner's dilemma', *The dynamics of norms*: 1–16.

Agar, Michael. 1997a. 'Advancing the Art of Simulation in the Social Sciences.' in Rosaria Conte, Rainer Hegselmann and Pietro Terna (eds.), *Simulating Social Phenomena* (Springer Berlin Heidelberg: Berlin, Heidelberg).

Agar, Michael. 1997b. *The complexity of cooperation: agent-based models of competition and collaboration* (Princeton University Press: Princeton, N.J.).

Akimov, Vladimir, and Michail Soutchanski. 1994. 'Automata simulation of n-person social dilemma games', *Journal of Conflict Resolution*, 38: 138–48.

Axelrod, Robert. 1980. 'Effective Choice in the Prisoner's Dilemma', *Journal of Conflict Resolution*, 24: 3–25.

Axelrod, R. (1986). "An Evolutionary Approach to Norms." *The American Political Science Review* 80(4): 1095–1111.

Axelrod, Robert M. 2006. *The evolution of cooperation* (Basic books).

Axtell, Robert, Robert Axelrod, Joshua M. Epstein, and Michael D. Cohen. 1996. 'Aligning simulation models: A case study and results', *Computational & Mathematical Organization Theory*, 1: 123–41.

Barabási, Albert-László, and Réka Albert. 1999. 'Emergence of scaling in random networks', *science*, 286: 509–12.

Barreteau, Olivier, Martine Antona, Patrick D'Aquino, Sigrid Aubert, Stanislas Boissau, François Bousquet, William's Daré, Michel Etienne, Christophe Le Page, and Raphaël Mathevet. 2003a. 'Our companion modelling approach', *Journal of Artificial Societies and Social Simulation*, 6.

Barreteau, Olivier, Patrice Garin, Alexandre Dumontier, Géraldine Abrami, and Flavie Cernesson. 2003b. 'Agent-based facilitation of water allocation: Case study in the Drome River Valley', *Group Decision and Negotiation*, 12: 441–61.

Barreteau, Olivier, Christophe Le Page, and Pascal Perez. 2007. 'Contribution of simulation and gaming to natural resource management issues: an introduction', *Simulation & Gaming*, 38: 185–94.

Bennett, D Scott. 2008. 'Governments, civilians, and the evolution of insurgency: Modeling the early dynamics of insurgencies', *Journal of Artificial Societies and Social Simulation*, 11: 7.

Bhavnani, Ravi. 2003. 'Adaptive agents, political institutions and civic traditions in modern Italy', *Journal of Artificial Societies and Social Simulation*, 6.

Bhavnani, Ravi, and David Backer. 2000. 'Localized Ethnic Conflict and Genocide', *Journal of Conflict Resolution*, 44: 283–306.

Bianchi, Nicola. 1989. 'System Dynamics as a Tool for Historians: The Role of Software.' in Peter M. Milling and Erich O. K. Zahn (eds.), *Computer-Based Management of Complex Systems: Proceedings of the 1989 International Conference of the System Dynamics Society, Stuttgart, July 10–14, 1989* (Springer Berlin Heidelberg: Berlin, Heidelberg).

Blalock, Hubert M. 1974. *Measurements in the Social Sciences: Theories and Strategies* (AldineTransaction).

Borero, Riccardo, and Flaminio Squazzoni. 2005. 'Does empirical embeddedness matter? Methodological issues on agent-based models for analytical social science', *Journal of Artificial Societies and Social Simulation*, 8.

Bousquet, Antoine, and Robert Geyer. 2011. 'Introduction: complexity and the international arena', *Cambridge Review of International Affairs*, 24: 1–3.

Bousquet, François. 2005. *Companion modeling and multi-agent systems for integrated natural resource management in Asia* (Int. Rice Res. Inst.).

Castro, José. 1999. 'The origin of institutions: socio-economic processes, choice, norms and conventions', *Journal of Artificial Societies and Social Simulation*, 2.

Cederman, Lars-Erik. 1997. *Emergent actors in world politics: How states and nations develop and dissolve* (Princeton, NJ).

Chadwick, Richard W. 2000. 'Global Modeling: Origins, Assessment, and Alternative Futures', *Simulation & Gaming*, 31: 50–73.

Clogg, Clifford C. 1992. 'The Impact of Sociological Methodology on Statistical Methodology', *Statistical Science*, 7: 183–96.

Dilaver, Ozge. 2015. 'From Participants to Agents: Grounded Simulation as a Mixed-Method Research Design', *Journal of Artificial Societies and Social Simulation*, 18: 15.

Doran, Jim. 2001. 'Intervening to Achieve Co-operative Ecosystem Management: Towards and Agent Based Model', *Journal of Artificial Societies and Social Simulation*, 4.

Durkheim, Emile. 1897. *Le Suicide. Etude de sociologie* (Paris).

Durkheim, Emile, and Steven Lukes. 1982. *The rules of sociological method* (Free Press: New York).

Earnest, David C. 2015. *Massively Parallel Globalization: Explorations in Self-Organization and World Politics* (SUNY Press).

Epstein, Joshua M. 2006. *Generative social science: studies in agent-based computational modeling* (Princeton University Press: Princeton).

Epstein, Joshua M. 2008. 'Why Model?', *Journal of Artificial Societies and Social Simulation*, 11: 12.

Epstein, Joshua M. 2013. *Agent zero: toward neurocognitive foundations for generative social science* (Princeton University Press: Princeton and Oxford).

Epstein, Joshua M., and Robert Axtell. 1996. *Growing artificial societies: Social science from the bottom up* (Brookings Institution Press: Washington, D.C.).

Geller, Armando, and Scott Moss. 2008. 'Growing qawm: An evidence-driven declarative model of Afghan power structures', *Advances in Complex Systems*, 11: 321–35.

Gibbons, Robert. 1992. *Game theory for applied economists* (Princeton University Press: Princeton, N.J.).

Gilbert, G. Nigel. 2008. *Agent-based models* (Sage Publications: Los Angeles).

Gilbert, G. Nigel, and Klaus G. Troitzsch. 2005. *Simulation for the Social Scientist* (Open University Press: Berkshire, England).

Gurung, Tayan Raj, Francois Bousquet, and Guy Trebuil. 2006. 'Companion modeling, conflict resolution, and institution building: sharing irrigation water in the Lingmuteychu Watershed, Bhutan', *Ecology and Society*, 11.

Harsanyi, John C. 1967. 'Games with Incomplete Information Played by "Bayesian" Players, I–III Part I. The Basic Model', *Management Science*, 14: 159–82.

Hughes, Barry B. 1999. 'The International Futures (IFs) Modeling Project', *Simulation & Gaming*, 30: 304–26.

Jacobson, Michael J., and Uri Wilensky. 2006. 'Complex Systems in Education: Scientific and Educational Importance and Implications for the Learning Sciences', *Journal of the Learning Sciences*, 15: 11–34.

Janssen, Marco A., and Elinor Ostrom. 2006. 'Empirically Based, Agent-based models', *Ecology & Society*, 11.

Kahler, Miles. 1992. 'Multilateralism with small and large numbers', *International Organization*, 46: 681–708.

Kuznar, Lawrence A. 2006. 'High-Fidelity Computational Social Science in Anthropology', *Social Science Computer Review*, 24: 15–29.

Le Page, C, and P Bommel. 2005. 'A methodology for building agent-based simulations of common-pool resources management: from a conceptual model designed with UML to its implementation in CORMAS', *Companion Modeling and Multi-Agent Systems for Integrated Natural Resource Management in Asia*: 327–49.

Lerner, Reuven M., Sharona T. Levy, and Uri Wilensky. 2010. "Connected Modeling: Design and analysis of the Modeling Commons." In *Chais Conference on Instructional Technologies Research 2010: Learning in the Technological Era*. Raanana, Israel.

Liebrand, Wim BG. 1984. 'The effect of social motives, communication and group size on behaviour in an N-person multi-stage mixed-motive game', *European journal of social psychology*, 14: 239–64.

Lustick, Ian, Dan Miodownik, and Roy Eidelson. 2004. 'Secessionism in Multicultural States: Does Sharing Power Prevent or Encourage It?', *American Political Science Review*, 98.

Lustick, Ian S. 2000. 'Agent-based modeling of collective identity: testing constructivist theory', *Journal of Artificial Societies and Social Simulation*, 3.

Lustick, Ian S. 2011. 'Secession of the Center: A Virtual Probe of the Prospects for Punjabi Secessionism in Pakistan and the Secession of Punjabistan', *Journal of Artificial Societies and Social Simulation*, 14.

Macy, Michael W., and Robert Willer. 2002. 'From Factors to Actors: Computational Sociology and Agent-Based Modeling', *Annual Review of Sociology*, 28: 143–66.

Malerba, Franco, Richard Nelson, Luigi Orsenigo, and Sidney Winter. 2001. 'History-Friendly models: An overview of the case of the Computer Industry', *Journal of Artificial Societies and Social Simulation*, 4.

Matsushima, Masanao, and Takashi Ikegami. 1998. 'Evolution of strategies in the three-person iterated prisoner's dilemma game', *Journal of theoretical biology*, 195: 53–67.

Miller, John H. 1998. 'Active Nonlinear Tests (ANTs) of Complex Simulation Models', *Management Science*, 44.

Miodownik, Dan, and Britt Cartrite. 2010. 'Does Political Decentralization Exacerbate or Ameliorate Ethnopolitical Mobilization? A Test of Contesting Propositions', *Political Research Quarterly*, 63: 731–46.

Mitchell, Melanie. 2009. *Complexity: A guided tour* (Oxford University Press: New York).

Neumann, Martin. 2015. 'Grounded Simulation', *Journal of Artificial Societies and Social Simulation*, 18: 9.

Nowak, Martin A, and Robert M May. 1992. 'Evolutionary games and spatial chaos', *Nature*, 359: 826–29.

O'Rand, Angela M. 1992. 'Mathematizing social science in the 1950s: The early development and diffusion of game theory', *Toward a history of game theory*, 24: 177.

Ohtsuki, Hisashi, Christoph Hauert, Erez Lieberman, and Martin A Nowak. 2006. 'A simple rule for the evolution of cooperation on graphs and social networks', *Nature*, 441: 502–05.

Olson, Mancur. 2009. *The logic of collective action* (Harvard University Press).

Ormerod, Paul, and Bridget Rosewell. 2009. 'Validation and Verification of Agent-Based Models in the Social Sciences.' in Flaminio Squazzoni (ed.), *Epistemological Aspects of Computer Simulation in the Social Sciences* (Springer Berlin Heidelberg).

Ouml, Birnur zbaş, Ouml, zg, uuml, Onur n, and Yaman Barlas. 2014. 'Modeling and Simulation of the Endogenous Dynamics of Housing Market Cycles', *Journal of Artificial Societies and Social Simulation*, 17: 19.

Pahre, Robert. 1994. 'Multilateral cooperation in an iterated prisoner's dilemma', *Journal of Conflict Resolution*, 38: 326–52.

Picardi, Anthony C. 1976. 'Practical and ethical issues of development in traditional societies: Insights from a system dynamics study in pastoral West Africa', *SIMULATION*, 26: 1–9.

Polhill, J. Gary, Lee-Ann Sutherland, and Nicholas M. Gotts. 2010. 'Using Qualitative Evidence to Enhance an Agent-Based Modelling System for Studying Land Use Change', 13: 10.

Power, Conrad. 2009. 'A spatial agent-based model of N-person prisoner's dilemma cooperation in a socio-geographic community', *Journal of Artificial Societies and Social Simulation*, 12: 8.

Rasmussen, Laura Vang, Kjeld Rasmussen, Anette Reenberg, and Simon Proud. 2012. 'A system dynamics approach to land use changes in agro-pastoral systems on the desert margins of Sahel', *Agricultural Systems*, 107: 56–64.

Rousseau, David, and A. Maurits van der Veen. 2005. 'The Emergence of a Shared Identity: An Agent-Based Computer Simulation of Idea Diffusion', *The Journal of Conflict Resolution*, 49: 686–712.

Sandberg, Mikael. 2011. 'Soft Power, World System Dynamics, and Democratization: A Bass Model of Democracy Diffusion 1800–2000', *Journal of Artificial Societies and Social Simulation*, 14.

Santos, Francisco C, and Jorge M Pacheco. 2005. 'Scale-free networks provide a unifying framework for the emergence of cooperation', *Physical Review Letters*, 95: 098104.

Schelling, Thomas C. 1980. *The strategy of conflict* (Harvard university press).

Schelling, Thomas C. 1960. *The strategy of conflict* (Harvard University Press: Cambridge,).

Schelling, Thomas C. 1978. *Micromotives and Macrobehavior* (W. W. Norton & Company: New York, NY).
Skyrms, Brian, and Robin Pemantle. 2009. 'A dynamic model of social network formation.' in, *Adaptive networks* (Springer).
Snidal, Duncan. 1991a. 'International cooperation among relative gains maximizers', *International Studies Quarterly*, 35: 387–402.
Schelling, Thomas C. 1991b. 'Relative Gains and the Pattern of International Cooperation', *American Political Science Review*, 85: 701–26.
van Dijkum, C., J. van Mens-Verhulst, E. van Kuijk, and N. Lam. 2002. 'System Dynamic Experiments with Non-linearity and a Rate of Learning: the development of chronic fatigue complaints', *Journal of Artificial Societies and Social Simulation*, 5.
Vernon-Bido, Daniele, Erika Frydenlund, J. J. Padilla, and David C. Earnest. 2017. "Durable solutions and potential protraction: The syrian refugee case." In *Spring Simulation Multi-Conference 2017*. Virginia Beach, VA: Springer.
Von Neumann, John, and Oskar Morgenstern. 1944. *Theory of games and economic behavior* (Princeton university press: Princeton,).
Watts, Duncan J, and Steven H Strogatz. 1998. 'Collective dynamics of 'small-world'networks', *nature*, 393: 440–42.
Weaver, Warren. 1948. 'Science and Complexity', *American Scientist*, 36.
Wilensky, Uri, and Mitchel Resnick. 1999. 'Thinking in Levels: A Dynamic Systems Approach to Making Sense of the World', *Journal of Science Education and Technology*, 8: 3–19.
Wolfram, Stephen. 1985. "Complex systems theory." In. Princeton: The Institute for Advanced Study.
Wolstenholme, E. F., and R. G. Coyle. 1983. 'The Development of System Dynamics as a Methodology for System Description and Qualitative Analysis', *Journal of the Operational Research Society*, 34: 569–81.
Yamamoto, Kazuya. 2015. 'Mobilization, Flexibility of Identity, and Ethnic Cleavage', *Journal of Artificial Societies and Social Simulation*, 18: 8.
Yang, Lu, and Nigel Gilbert. 2008. 'Getting Away From Numbers: Using qualitative observation for agent-based modeling', *Advances in Complex Systems*, 11: 175–85.

Author Biographies

David C. Earnest, Ph.D., is Associate Dean for Research & Graduate Studies in the College of Arts & Letters, and Professor of Political Science and International Studies at Old Dominion University, Norfolk, Virginia, USA. As a scholar of globalization and the global political economy, his research investigates how people create institutions to manage complex interactions between economic, political, ecological, technological and physical systems. Dr. Earnest's publications and research have contributions to several transdisciplinary fields including globalization theory, complex systems theory, and computational social science. Dr. Earnest is author of three books: Massively Parallel Globalization: Explorations in Self-Organization and World Politics (SUNY Press 2015); Old Nations, New Voters: Nationalism, Transnationalism and Democracy in the Era of Global Migration (SUNY Press 2008) and On the Cutting Edge of Globalization (Rowman and Littlefield 2005). He has published findings in International Studies Quarterly (2008) and has written about the epistemology of computational methods in Complexity in World Politics (Neil Harrison, ed., SUNY Press 2006). He has also published in leading journals including World Politics, Foreign Policy, International Interactions, and Globalizations.

Erika Frydenlund, Ph.D., is a Research Assistant Professor at VMASC at Old Dominion University. Her research focuses on social justice and mobility, specifically refugees and economically vulnerable populations. Dr. Frydenlund participated as a graduate student at the Santa Fe Institute's Graduate Workshop in Computational Social Science Modeling and Complexity. She served as a program chair on the Annual Simulation Symposium (ANSS) for the Spring Simulation Multi-Conference 2017, and a member of the Head of Projects team for the Emerging Scholars and Practitioners of Migration Issues (ESPMI). She served for five years on the local community advisory board for refugee resettlement in Hampton Roads. Dr. Frydenlund's research predominantly uses agent-based modeling to model social justice issues.

Chapter 12
Simulation-Based Enterprise Management

Model Driven from Business Process to Simulation

Gregory Zacharewicz, Amir Pirayesh-Neghab, Marco Seregni, Yves Ducq and Guy Doumeingts

Abstract Industrial enterprises are gradually integrating Modeling & Simulation (M&S) approaches to support their management processes and to keep themselves competitive in the market by handling and connecting more efficiently their key information. On the one hand, several modeling solutions exist, with different views or abstraction levels, which are not always compatible; on the other hand, the usage of simulation for enterprise management should be aligned with the nature of decision-making. This hinders the choice of an adapted M&S solution. To facilitate the resolution of this issue, this chapter mainly proposes to apply Model Driven Service Engineering Architecture (MDSEA), which guides the usage of M&S for enterprise management at business/technical levels or with static/dynamic points of view. In its first part, the chapter focuses on different state-of-the-art elements (e.g., Enterprise Modeling, Discrete Event Simulation, etc.) which support the development of a *simulation-aided decision making cycle* for enterprise management. Simulation models involved in this cycle can be gradually created from transformation of high-level or static models. An example of such transformation is described in the second part of this chapter. The objective is to move from BPMN 2.0 (Business Process Model and Notation) to DEVS (Discrete EVent Specification) which is a simulation-ready language. The second part ends by presenting a use-case and the implemented open-source software, called Service Lifecycle Management Tool Box (SLMToolBox). The chapter is concluded by discussing the propositions and the perspectives, particularly simulation of decision models for enterprise management.

G. Zacharewicz (✉) · Y. Ducq · G. Doumeingts
Laboratory IMS CNRS Production Engineering Team, Talence Cedex, France
e-mail: gregory.zacharewicz@ims-bordeaux.fr

A. Pirayesh-Neghab · G. Doumeingts
INTEROP-VLab, Brussels, Belgium

M. Seregni
Department of Management, Economics and Industrial Engineering,
Politecnico Di Milano, Milan, Italy

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_12

Keywords Simulation · Enterprise management · M&S · Enterprise modeling · Business process · MDE · MDI · MDA · MDSEA · GRAI · Hierarchical decomposition · Decision aid · DEVS · BPMN · EA* · SLMTool box · Model transformation · ATL · Interoperability · System

12.1 Introduction

To remain competitive, a company must differentiate itself from other competitors based on higher value propositions (products or services) or performant business processes. Since improving the product's performance can reach some limits, one open solution is to improve the enterprise system, redefine its processes, and share more information (considered as additional services) with customers and suppliers; the so-called servitization. Indeed, this is the role of enterprise management "*to design, control and improve the business processes in order to adapt to the changing business environment to cope with innovations, mutations of customers' expectations and increasing competition*" (Burlton 2001; Sienou et al. 2007). Therefore, management requires a proper understanding of the enterprise in order to rapidly and precisely evaluate its performance.

Facing the above needs, enterprise system complexity, particularly in the manufacturing context, has been mentioned as a real challenge since a long time (Wiendahl and Scholtissek 1994). ElMaraghy et al. have reviewed this subject, its factors and sources in the design process, products, manufacturing, and business (ElMaraghy et al. 2012). The complexity can be due to the following:

– the variety of the components' nature; human, social, technical, economic, organizational, etc.
– the numerous interactions of these components; internally, between enterprise resources and externally with the environment, customers, suppliers, and competitors.
– automation and new data exchange technologies (ElMaraghy et al. 2012) and developments in the global market.
– the excessive data and information.

Complexity is usually followed by the introduction of uncertainty and risk to the enterprise systems through creation of several internal and external factors (e.g., demand, market share, supply rate …) which affect the performances and the final output of the system.

In such an environment, Enterprise Modeling techniques can be applied as preliminary support for management by simplifying the representation of the processes. Nevertheless, models provide solely a static abstraction of enterprise system. To go further, simulation can be necessary for providing assessments of the system performance regarding dynamicity and behavior, for both existing system and the one to be developed (Pirayesh-Neghab et al. 2011; Bruzzone et al. 2000). Therefore, joint modeling and simulation (M&S) is a necessity for enterprise management.

Considering the plethora of existing M&S solutions, one current approach consists of proposing platforms or toolboxes gathering a set of ad hoc solutions. This might help identifying solutions for a specific and clearly defined problem. Nevertheless, modeling expert is faced with issues such as lack of conceptual or technical interoperability when it is required to transpose models and simulations when moving from solution to another one. Also the model contents and simulation results are still difficult to exploit with another level of abstraction. It is mainly due to lack of methodological connection between M&S solutions. For example, (Zacharewicz et al. 2016) has stated that it remains difficult to pave the way from conceptual models to executable models (simulation).

The problem of selecting the adapted M&S is even more radical considering the different levels of decision-making (i.e., strategic, tactical, and operational) in enterprise management (Doumeingts et al. 1998). Thus, in order to support enterprise management, it is important to study the way to model and decompose decisions, to connect the decisions to adapted simulations, and to aggregate the simulation results according to the decision level. Another difficulty for selecting the adapted M&S simulation is the complexity of decision-making process due to multiplicity of information, which information (e.g., performance indicator) should be taken into account for simulation; which simulation result can support the decision-making; and finally, how the results can be aggregated according to the decision-making level.

Therefore, the usage of M&S solutions should be, on the one hand, followed by a structural architecture and a methodology covering different points of view (e.g., business or technical, static, or dynamic) and, on the other hand, capable of considering and relating to the different layers of enterprise management and decision-making (e.g., from machine control to factory management in a manufacturing system). In order to select the appropriate M&S solution, a model-driven architecture can be adopted. These architectures are mainly proposed in the frame of Model Driven Engineering. For instance, Model Driven Service Engineering Architecture (MDSEA) (MSEE Book 2014), supporting the resource management to improve the enterprise performance, is discussed in this chapter. In this architecture, simulation is considered as an aid for the decision cycle. Several types of simulation can be executed such as product, information workflow, or process simulation. In MDSEA, the informational process simulation is started from higher level process models which gradually integrate technical aspects to become "simulation-ready".

After this introductory section, an overview of research literature is addressed while briefly presenting enterprise modeling, simulation, and model-driven concepts and methods. Then a global architecture situating simulation as decision aid is presented. Within this architecture, an example of process simulation focused on model transformation from BPMN 2.0 to DEVS is also introduced including the illustration on the use-case. Finally, the perspectives of this work are proposed, particularly for the transformation of decision processes to simulation models.

12.2 Model-Driven Enterprise Management

This section presents the concept of Model Driven Enterprise Management which is founded on the methods of MDE and Enterprise Modeling.

12.2.1 MDE

MDE (Schmidt 2006) is a system engineering approach that uses the capabilities of conceptual representations of a system independently of computer technologies. MDE adopts models and languages in order to describe both the problem posed (need) and its solution. Then it goes smoothly to concrete solution.

As a structured method in the frame of MDE, MDA (Model Driven Architecture) can be mentioned. This method, defined and adopted by the OMG (Object Management Group) in 2001, then updated in 2003 (MDA 2003), is designed to promote the use of models and their transformations to consider and implement different systems. It is a four-level architecture guiding the passage from generic to specific models of a software product.

Based on MDA and in the frame of Task Group 2 (TG2) of INTEROP-NoE, the approach "Model Driven Interoperability" (MDI) considers interoperability problems from enterprise models level instead of only at the technical or coding levels (Bourey et al. 2007). The main goal of MDI, based on modeling, is to allow a complete follow, through model transformation, from expressing interoperability requirements (determination of barriers) to coding of a solution. This approach provides a greater flexibility, thanks to the automation of these transformations.

12.2.2 MDSEA

MDSEA is an engineering architecture which has been recently proposed in the servitization context based on MDI (Ducq et al. 2014). This architecture was developed in the frame of MSEE project (MSEE book 2014).

The main goal of MDSEA is to model enterprise system and support the development of its major components in three domains: Information Technology (IT), Human/Organization, and Physical Means (i.e., machine or any physical tool) (see Fig. 12.1). One of the originalities of the approach lays on these passages:

- from BSM (Business Service Model) level, as the high-level abstraction adapted to the business point of view, till TIM (Technology Independent Model) level, as the technical point of view regardless of the technology choice.
- from TIM level to TSM (Technology Specific Model) level, as the detailed technical level resulting in the final development of components.

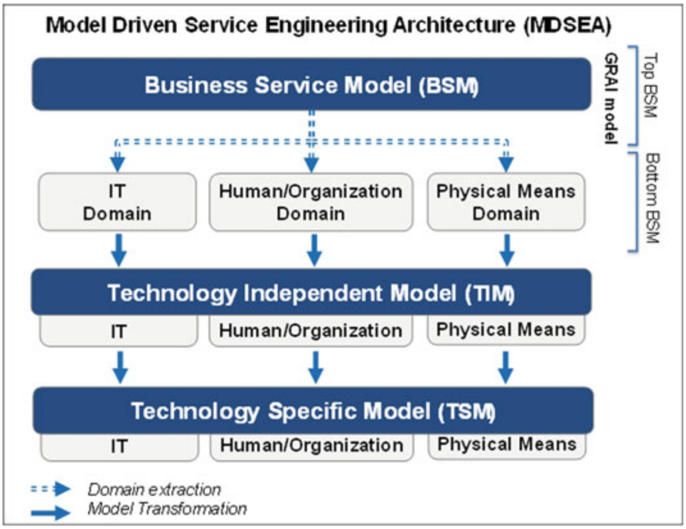


Fig. 12.1 Model Driven Service Engineering Architecture (MDSEA)

The above passages can be supported by Model Transformation which is based on a sequence of Top-Down and Bottom-Up iterations; the more we progress toward the development of the solution, the more complementary and detailed information is necessary.

MDSEA starts from an integrated and high-level modeling at BSM level (Top BSM) which consists of several integrated enterprise models elaborated based on GRAI (Graph with Results and Activities Interrelated) methodology and its formalisms. From BSM models, it is then necessary to extract the elements which will allow describing each of the three domains in order to create three main categories of resources (bottom BSM) (see "domain extraction" in Fig. 12.1). It is worth mentioning that even though these domains are focused on specific type of resource, they are not completely independent and might be overlapping. For each domain, the model at TIM level is created by a transformation of the BSM bottom using Model Transformation.

For instance, models can be elaborated with Extended Actigram Star (EA*), which can be mentioned as a high-level business process modeling languages used at top BSM level. Concepts related to IT domain, which are necessary for the development of IT resources, are first extracted from EA* models. Then, they are transformed to corresponding concepts of BPMN model, at TIM level (see (a) in Fig. 12.1).

12.2.3 Enterprise Modeling

Enterprise Modeling (EM) allows the representation of enterprise with concepts which aim at describing the strategy, the processes, the functionalities, the organization, the decisional structure, the evolution in time, the relationships with the environment (e.g., with customers and suppliers), etc. EM, through elaboration of enterprise models, supports the understanding of an enterprise system with the objective of analyzing and improving its performance. These methods can be applied not only on industrial enterprises, but also in services and public administrations such as hospitals and teaching institutes.

In order to design the enterprise model, ad hoc conceptualization or abstraction methods could be applied. Reference or standard methods could also support this task by providing a common view among different industrial branches, clarifying the current trends, key dimensions, and layers of the system.

According to a survey on the tools for system modeling, by extension enterprise modeling (Kettinger et al. 1997), there are lots of modeling languages and tools available which are capable of different aspects of a system. Kettinger et al. listed over 100 tools.

In manufacturing context, in order to be able to improve the competitiveness in the 70s, United States Department of Defense (DoD) proposed to use Enterprise Modeling Techniques (EMT) for describing a manufacturing system according to its various aspects (i.e., activities, processes, information, and simulation) (Savage 1996); the pioneer was ICAM project from which IDEF (Integration DEFinition: IDEF0 … IDEFx) modeling method was born (Doumeingts and Ducq 2001). Since this time, several concepts, methods, and tools have been developed such as CIMOSA (Computer Integrated Manufacturing Open System Architecture), GRAI model, GIM (GRAI Integrated Modeling), EIM (Enterprise Integration Modeling), ARIS (Architecture of Integrated Information Systems), etc. (Doumeingts and Ducq 2001; Chen et al. 2008).

Another approach for representation of enterprise structure is Enterprise Architecture which is more ICT-oriented. Such architectures can be developed to provide an abstract view of the ICT structure. A system architecture is defined as "*a conceptual model of a system together with models derived from it that represent (1) different viewpoints defined as views on top of the conceptual model, (2) facets or concerns of the system in dependence on the scope and abstraction level of various stakeholders, (3) restrictions for the deployment of the system and description of the quality warranties of the system, and (4) embedding into other systems*" (Jaakkola and Thalheim 2011). A survey of the viewpoints from the most popular enterprise architectures (e.g., Zachman, Sagace Matrix, DoDaf 1.5, Pera, MDA, AFIS, and Pahl & Beitz) has been performed in Benkamouna et al. (2014).

In GRAI method (Doumeingts 1984; Chen and Doumeingts 1996), a system or particularly an enterprise system is decomposed, into three subsystems according to the System Theory (Le Moigne 1977), Management Decision (Simon 1969), and

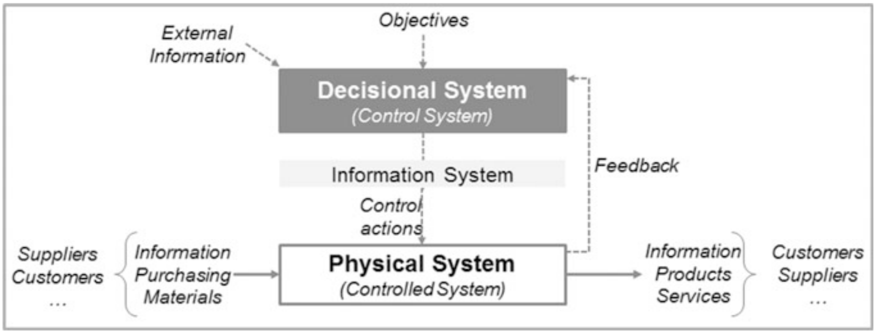


Fig. 12.2 GRAI model for integrated BSM model (top BSM) of MDSEA

Theory of Hierarchical Multilevel Systems (Mesarovic et al. 1970). A brief description of these subsystems is given below and shown in Fig. 12.2:

– The controlled subsystem (physical subsystem) transforms the inputs (materials and information) into outputs (new information, products or services) to be mainly delivered to the customers.
– The control subsystem (decisional subsystem) manages the physical subsystem based on the objectives of the global system (e.g., enterprise system) and the feedback information in order to delivers actions or adjustments.
– The information subsystem includes information from the physical subsystem and from the customers, suppliers, and other stakeholders (external environment).

12.2.3.1 Decisional Modeling

Regarding the representation of decisional subsystem, hierarchical decomposition and aggregation of information is a real necessity and challenge, particularly in manufacturing as complex enterprise system (see Fig. 12.3). This decomposition describes the different decision-making level (i.e., strategic, tactical, and operational). Furthermore, the decomposition highlights the required information to be provided by decision aids (e.g., simulation).

For a decision-maker at operational level of enterprise system, the scope and time span of decision-making is small (e.g., daily scheduling of a machine) and usually detailed information (e.g., machine capacity or machining time) are required which correspond to the reality. However, a decision-maker in charge of tactical or strategic levels is in charge of decisions (e.g., annual production planning) which covers numerous information with larger scope and time span (e.g., annual material requirements).

Another reason for a proper decisional modeling is cognitive limitation; the quantity of information that a decision-maker is able to process in a unit of time should be limited (Doumeingts et al. 1998).

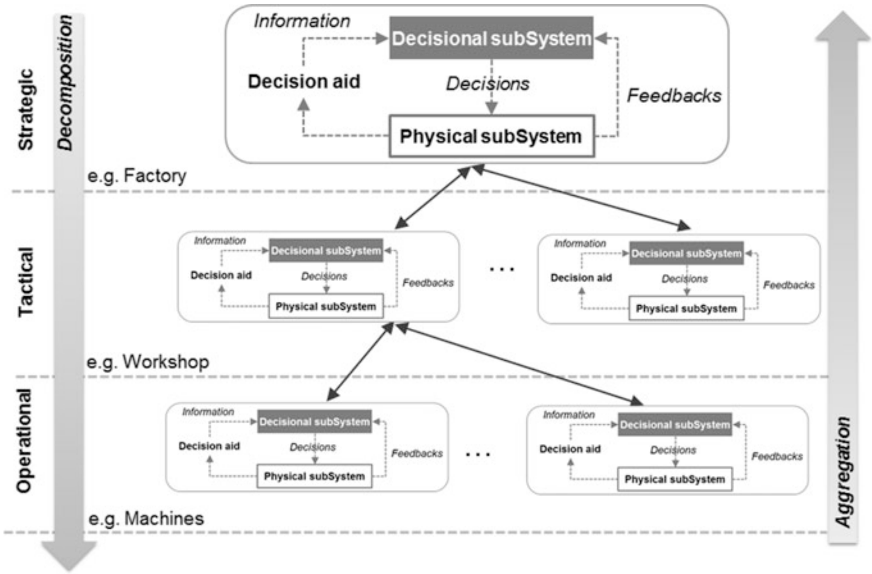


Fig. 12.3 Decomposition/aggregation of decisional subsystem

In GRAI model, the decisional modeling is performed using the GRAI Grid which takes into account the enterprise layers according the decision complexity. Such layers are based on the concept of horizon, the timespan of interest for the decision, and period, the timespan for (re)-evaluating a decision to find deviations from the expected pattern. "*The GRAI Grid does not aim at the detailed modeling of information processes [for decision making], but puts into a prominent position the identification of those points where decisions are made in order to manage a system*" (Doumeingts et al. 1998). The characteristic of the GRAI Grid is based on Decision Management (Simon 1969) and Theory of Hierarchical Multilevel Systems (Mesarovic et al. 1970). It should be mentioned that the GRAI Grid is coupled with GRAI Nets which provide a more detailed perspective about activities forming the decision-making process and the relationships among them.

Considering the dichotomy of decisional/physical subsystems in an enterprise (see Fig. 12.3), each decisional level is in charge of controlling a specific part of the physical subsystem. Therefore, the decomposition/aggregation of decisional modeling should be aligned with the physical subsystem modeling (e.g., process models). Indeed, the model of the physical subsystem is different at each level of decision using aggregated information where criteria of aggregation must be properly determined.

12.2.3.2 Business Process Modeling

As an example of methods for representing the physical and information subsystems (see GARI Model in Sect. 12.2.3), Business Process Modeling (BPM) can be mentioned (Cardoso et al. 2012). BPM results in a representation of an organization's business processes to be analyzed and improved (Weske 2007).

The world of business processes has changed dramatically over the past few years. Processes can be coordinated from behind, within and over organizations' natural boundaries. A business process now spans multiple participants and coordination can be complex. Business process models can help business actors to handle the problems of heterogeneity, complexity, and flexibility in layered operational Enterprise Architectures and across the enterprise knowledge spaces of network life cycles. Business process is a structured, measured set of activities designed to produce a specific output (product or service) for a particular customer or market (MSEE D15.2 2012). A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action.

One of the most commonly used approaches for BPM is Object-Oriented (O-O). This approach is based on a set of object classes which substitute the behavior of real process components (Anglani et al. 2002). Through O-O modeling the real process components are categorized into three different types of flows: materials, information, and decisions (Chen and Lu 1997). Chen and Lu presented an O-O oriented methodology using the Petri nets, the ERD (Entity Relationship Diagram), and the IDEF0 (Integrated DEFinition language for functional modeling) (Chen and Lu 1997).

Another commonly used languages is called BPMN (Business Process Model Notation) which is a standard notation for BPM. BPMN is supported by the Object Management Group (OMG 2003). Its objective is to provide a framework to describe a process in a way that is common to all users, irrespectively of the tool used. The tool is of course supposed to support the standard that is currently BPMN 2.0.

In MDSEA architecture (see Fig. 12.1), BPMN 2.0 is applied at TIM level for the IT domain. However, Business Process modeling is ideally started from a higher abstraction level (at BSM top level) using Extended Actigram Star (EA*). The main objective is to provide a common and simple modeling notation to the business user through an accessible syntax (MSEE D15.2 2012). Rather than EA*, other languages such as Archimate Business Level modeling can be also applied for this purpose. However, considering the focus of MDSEA on the management, development, and interoperability of different categories of resources, we believe that EA* might be a better choice since on the one hand, it distinguished the category of the resources and on the other hand, it clarifies the contents of flows (e.g., information, physical, sequential) in the process model.

In comparison to BPMN, EA* might reduce more the gap between the ideation, from user point of view, and the design of business process. EA* models can be transformed to BPMN model (MSEE D15.2 2012). In this research work, we directly apply BPMN models which are enriched after obtaining their skeleton from EA* models. The drawback of BPMs is due to the fact that it provides only a suitable static view which is missing the temporal dimension to express output performance such as an expected cost or a desired duration. In detail, the impact of correct or incorrect behavior of complex models over time is not clearly visible using static view. This issue can be solved by running a business process simulation for analyzing and understanding the business process model according to its dynamic.

BPMN is frequently associated to Business Process Execution Language (BPEL) (Thatte et al. 2003), that is a programming language for running business processes. Nevertheless, BPEL is intended to execution rather than simulation and it is not associated to clear execution semantics. The notions of states, dynamics remains open to the interpretation of the modelers potentially subjective so difficult to reuse and compare. Formal modeling theories can overcome this limitation by extending the credibility of the simulation models. It provides a sound ground for comparing the results and opening interoperability with different simulation platforms.

12.2.4 Simulation for Enterprise Management

Simulation solutions are designed according to different enterprise needs such as tracking performance indicators or providing information about the real behavior of products, services or processes, in a didactical and pedagogical way to support decision-making.

Manufacturers, for instance, can shorten time needed to develop new products/services and to (re)-engineer business process-related activities. At the same time, simulation techniques are expected support them in gaining new customers, lowering costs, improve business processes, increase customer satisfaction, and getting access to know-how. Thus, simulation techniques are now orienting toward customized applications in the form of Software-as-a-Service (SaaS), based on visual modeling tools to interactively create a model of the reality of interest for the specific user.

Simulation, and by extension M&S, as a service can be expected as a pre-validation on a perspective scenario of enterprise management. It is run on models of future enterprise management, and this is done by anticipation using probable and credible set of data. The simulation requires significant amount of data and resources to run the process and to consider the behavior of the process activities. Cloud-based approaches and High-Performance Computing (HPC) give more and more support to enhance these simulations. Nevertheless, the collaboration between HPC cloud infrastructure providers and Engineering Manufacturing users is often difficult (confidentiality, interoperability issues) and requires long times and huge efforts. So the remaining question is how to mobilize companies and particularly SMEs and midcaps to benefit from simulation digitization and HPC facilities to improve their competitiveness?

Several European projects (e.g., CloudSME, Fortissimo2, etc.) have been devoted to cope with the aforementioned challenge. For example, the project CloudSME (www.cloudsme.eu) supports SMEs to utilize. SMEs require the development of models involving their activities. Therefore, CloudSME proposes models that are simulated numerically by using either continuous or discrete event simulation techniques. The modeling and simulation processes take advantage of visual modeling tools to interactively create a model of the reality of interest. Fortissimo and Fortissimo2 (www.fortissimo-project.eu/) are collaborative projects that enable European SMEs to be more competitive globally through the use of simulation services running on HPC cloud infrastructures. The Fortissimo project (through its Marketplace) provides a number of approaches required by SMEs to find solutions to their challenges, which are mainly on-demand access to advanced simulation and modeling resources, and access to state-of-the-art HPC facilities, leading to a reduced computation time.

SMEs involved with the solutions provided by the two presented projects have benefited of a reduction of the design costs, thanks to the use of a set of simulation software ported to HPC system available through the cloud. Especially it appeared that for the use of a federation of heterogeneous simulations a workflow or process model was required to cope with the different solutions and services. Nevertheless, the modeling of this workflow and its transformation to simulation model is not simple and even explicit enough to be given directly to the usage of SME. Discrete event simulation can be a solution to model formally, and then orchestrate such a workflow of M&S in the aim to drive anticipation of enterprise management scenarios.

12.2.4.1 Discrete Event Simulation

Discrete Event Simulation (DES), which is a frequently used method in process M&S, significantly facilitates the enterprise management process definition and validation (Pirayesh et al. 2011). Semini et al. (2006) performed a literature survey on use of discrete event simulation in real-world manufacturing & logistics decision-making in 2006. According to this survey there are several reasons why a simulation study can support manufacturing and logistics decision-making:

- provides better understanding of the real system and its behavior.
- reveals previously hidden relationships.
- performs a systematic analysis of the situation.
- facilitates communication and provide a basis for discussions.
- allows the decision-maker to test the influence of different alternative scenarios without having to make changes in the real system.

Another review on simulation in manufacturing and business was presented by Jahangirian et al. in 2010. In this survey simulation is recognized as the second most widely used technique in the field of operation management and it has been applied in areas such as manufacturing, services, defense, healthcare, and public services. In real-world applications there are factors to be considered in selecting a proper simulation technique. Jahangirian et al. suggest that in case of dealing with different layers of decision-making within a system, a better understanding will be needed of the relationship between the different layers of organizations and of the way to connect simulation tools that relate to each layer in order to deal with the system as a whole (Jahangirian et al. 2010).

Simulation has been a widely used tool for manufacturing system design and analysis. It has proven to be an extremely useful analysis tool, and many hundreds of articles, papers, books, and conferences have focused directly on the topic. Smith presented a classification of a subset of these publications and the researches and applications that underlie these publications (Smith 2003).

As an example, in the nuclear industry, Monte Carlo simulation is proper for the study of system availability/reliability and component importance. Monte Carlo simulation involves no complex mathematical analysis and is preferred to deterministic methods which are difficult to solve specially in case of large and complex systems (Wu 2008).

Zeigler has proposed since 1976 the Discrete EVent Specification (DEVS) (Zeigler et al. 2000) as an integrated formalism which enhances the model designing efficiency with unambiguous specification formalism and provides a methodology for execution process by means of an executable semantics. We have chosen DEVS as the simulation language for the reasons enounced previous in order to remove ambiguity and unify the M&S concepts.

12.2.4.2 Simulation Tools

There are numerous software products in the simulation field. Semini et al. identified the papers using DES software tool in a literature survey. For instance they reported in the use of several relevant application papers from the last decade of Winter Simulation Conference proceedings. Arena and Automod/Autosched were used most frequently, followed by Quest, ProModel, Sigma, and Extend (Semini et al. 2006). Also simulation tools are developed in the academic and/or open source context (NetLogo, MS4ME, VLE, etc.).

DEVS supporting tools deserve a particular attention since this language has been selected for targeting simulation within this chapter. The DEVS group standardization maintains on its website the updated list of most used DEVS tools known by the DEVS community (Wainer 2013). In Hamri and Zacharewicz (2012), the authors have given a brief description and comparison of popular tools.

ADEVS was the first DEVS tool developed. DEVSJAVA is a Java framework in which the kernel simulator is ADEVS. CD++ Builder is a DEVS modeling and simulation environment that integrates interesting features and facilities for the user.

Other DEVS tools are dedicated to specific areas. VLE is a C++ M&S framework that integrates heterogeneous models from different scientific fields. This integration is based on the agent paradigm. In addition, JDEVS is the Java implementation of a DEVS formal framework. It supports multi-modeling paradigms based on DEVS. It ensures the interoperability among the reused components. Also SIMSTUDIO can be considered, and it is focused on a simplified DEVS editor for DEVS non-Expert. The authors also investigate LSIS_DME that is focused on a graphical interface and code source generation in order to complete the model by complex Java functions.

At the end each DEVS editor is covering interesting aspects that complete basic DEVS facilities or propose different model views. Nevertheless, we found it difficult to import by the tool non DEVS models other than hard coded matching, i.e., the customization is limited. We suggest that the feeding by other model can be facilitated if following a Model-Driven approach, e.g., MDA. One core concept of MDA is the Meta Model that is required for model matching, an example of which has been proposed by Garredu et al. (2012).

12.2.5 Model Transformation

Considering the diversity of actors in product engineering, several heterogeneous standards or modeling languages, with different purposes, are applied. Therefore, treatments on models should be considered in order to achieve interoperability in model exchanges (Pirayesh et al. 2015). For this purpose, Model Transformation, defined as the process of converting a [Product or Process] Data model to another model of the latter (Miller and Mukerji 2003), can be mentioned. It is indeed considered as a common interoperability solution in MDE and is classified as a federative approach. A taxonomic classification of the various existing approaches for Model Transformation is proposed in Czarnecki and Helsen (2003). The authors also classify existing approaches of transformation as follows: direct-manipulation, relational, graph-transformation-based, structure-driven, and hybrid approaches.

Figure 12.4 shows the Model Transformation architecture, in the frame of Meta Object Facility (MOF) of OMG. In this architecture,

- A source model is transformed into a target model. These models, in M1 level, are in accordance (conformsTo) with their own meta-models of the M2 level, and meta-models are consistent with a single meta-meta-model (M3 level).
- Transformation is based on rules (mapping or projection). Indeed, the transformation is based on semantic and syntactic relations between models, which are developed by domain experts of these models.
- The transformation also requires a transformation language that implements the transformation rules. This language is itself conforming to level M3 (meta-meta-model) in the transformation architecture.

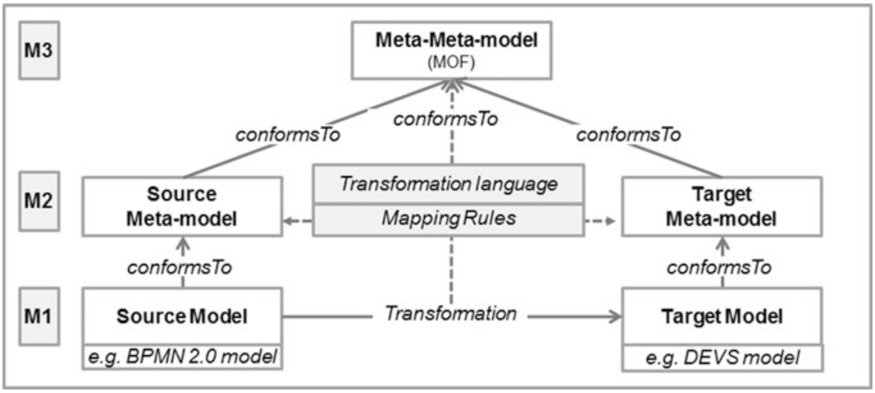


Fig. 12.4 Model transformation architecture and its key elements

As it is mentioned above, one of the key elements of Model Transformation is Data Abstraction. In abstraction, concepts and conceptual relationships are created as conceptual models (Lezoche et al. 2011). After abstraction and formalization, the mapping can be started as an operation that defines the transformation rules between a pair of representations [meta-models] (Shvaiko 2005). Alignment is the main part of mapping and the basic task of a mapping operator. A more complex mapping process is described in Bouquet et al. (2005).

Once the mapping is defined, a language is required for realizing the transformation. As an example of such language, ATLAS Transformation Language (ATL 2013) and MISTRAL (Kurtev and van den Berg 2005) are widely used in the context of MDE. ATL is a model transformation language specified as both a meta-model and a textual concrete syntax. It complies with MOF and provides a way to generate the target model from the source model for developers in MDE. ATL provides developers with a mean to specify the way to produce a number of target models from a set of source models.

Currently, there are several toolkits (e.g., Topcased), used as integrated plugins in the Eclipse platform, which support this language. These toolkits also allow the implementation of the transformation of XML documents as well as MOF or Ecore meta-models (Lu 2012).

In MDSEA architecture, according to the distinction of static and dynamic process modeling (Cardoso et al. 2012), the focus will be on a complementary step at TIM level. This step concerns transformation of static business process models (e.g., BPMN 2.0 model (OMG 2011), at TIM level, to a simulation model) able to analyze the behavior of the system (see Fig. 12.4). Based on the feedbacks provided by the simulation, the high-level process models (EA* model) can be modified. This cycle will continue till obtaining the most performant configuration of the system and its processes.

12.2.6 Simulation as Decision Aid for Enterprise Management

Considering the structure of enterprise system described in Sect. 12.2.3, its decisional subsystem receives information about the physical subsystem at various levels of management. Such information can be collected from the real operational processes, from the process models with different abstraction levels (the so-called digital twins), or from the simulation of the models. In such case, a simulation-aided decision cycle is formed (see Fig. 12.5). Here, we emphasize on the importance of a hierarchical structure, covering, and connecting different enterprise levels, for the use of simulation tool in a decision aid approach. In enterprise systems, it is not always possible to simulate the processes at the operational level due to the amount of information and the run time.

In the proposed approach, the first step (see (1) in Fig. 12.5) of the decision-making cycle is started by the decomposition of the decisions and the information [e.g., simulation needs and Performance Indicators (PIs)] supporting those decisions. This step can be performed using decisional modeling methods such as GRAI Grid (see Sect. 12.2.3). Then, the simulation solution should be selected according to the required information (see (2) in Fig. 12.5). For instance, in a manufacturing system, the decision at strategic level can be about the choice of suppliers. Therefore, in case of lacking historical data, the simulation solution is intended to provide an overall estimation of the consumed materials in a period of time.

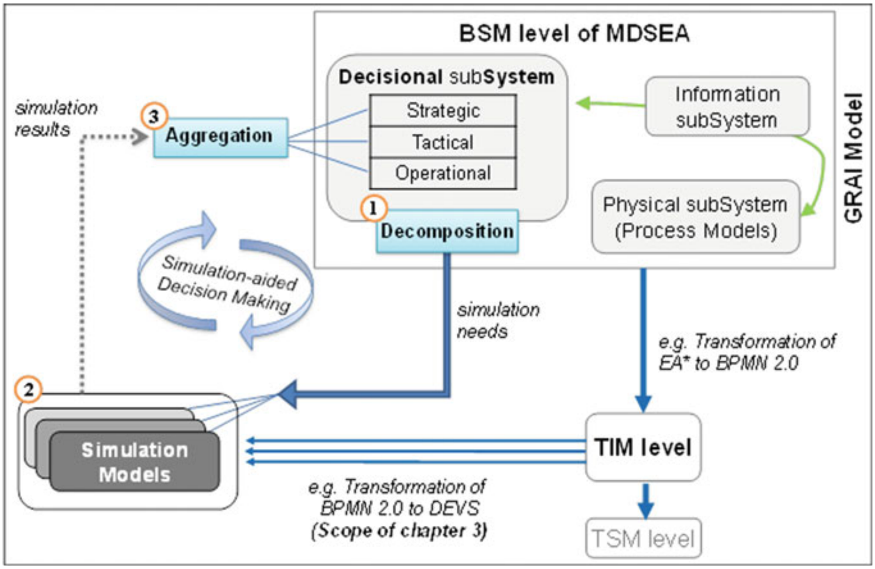


Fig. 12.5 Simulation-based decision aid for enterprise management in MDSEA

After the simulation, it is usually required to aggregate the results according to the same criteria of decomposition or enterprise layers (see (3) in Fig. 12.5). In the above example, the information about raw material consumption should be classified and aggregated based on annual consumption, category of product, overall cost of quality, etc.

The simulation models enabling the "*simulation-aided decision making cycle*" can be the result of transforming physical subsystem models (e.g., process models). The modeling work can be guided by Enterprise Modeling techniques.

As explained previously in Sect. 12.2.2, such modeling is often started at a high-level abstraction adapted to business point of view (BSM level of MDSEA). For instance, according to the decisions and objectives at strategic level of GRAI Grid a, process modeling can be performed using EA* language. Then, models can be transformed with two main purposes;

– enrichment of technical capacities: this is considered as a vertical transformation, from one level to another, in MDSEA architecture. EA* to BPMN is an example of transformation from BSM level to TIM level.
– provision of dynamic capacities: this is considered as a horizontal transformation, at the same level, in MDSEA. As an example of the latter, BPMN 2.0 to DEVS transformation can be mentioned which occurs at TIM level (from TIM-static to TIM-dynamic). This example is discussed in the following section of this chapter.

12.3 From Business Process to Simulation

Developing a high-level process model (business process model) before the development of the simulation model helps the recognition of the operation and also it is a time and cost saving act (Nethe and Stahlmann 1999). At design (or build time), it exists many process modeling languages. Yet, there are several reasons to choose BPMN among different formalisms. First, it is standardized by OMG and widespread in the industrial domain. Then, it can be generated from higher level languages such as EA*. Finally, it is associated to a set of execution languages. Moreover, it is important to adopt a federative formalism that can group the concepts of simulation and DES to be shared between different authors where the difference is the notations. For this purpose, DEVS language is selected which embraces a very large scope of domain.

This part presents the main principles of model transformation from Business Process to Simulation, based on the example of BPMN model to DEVS model, including the transformation architecture, DEVS meta-model, the mapping of concepts, and the implementation using a transformation language.

12.3.1 Background

In the context of BPMN to DEVS transformation, authors in Cetinkaya et al. (2012) and (Mittal and Risco Martin 2012) presented a Model Driven Development (MDD) framework for modeling and simulation (MDD4MS). In the frame of this framework they defined a model-to-model transformation from BPMN as a conceptual modeling language to DEVS as a simulation model specification. BPMN and DEVS meta-models were presented. In addition, a set of transformation rules were defined in order to transform BPMN models into DEVS models. According to these rules, some BPMN concepts (Pool, Lane, SubProcess) were mapped to DEVS coupled component, while Task, Event (Start, End, and Intermediate), and Gateway were mapped to DEVS atomic component.

Comparing the BPMN meta-model defined with the latest version of BPMN 2.0 meta-model (OMG 2011) we can conclude that several concepts are missing and thus were not transformed into their corresponding DEVS concept. Authors did not mention the different types of BPMN Tasks (User Task, Manual Task, Service Task…) and BPMN Intermediate Events (Message, Signal…) that can be mapped differently when transformed into DEVS concepts. The difference would be in the number of states forming each DEVS Atomic Model. Based on these remarks, the work presented in this chapter takes into consideration these points in an attempt to benefit from previous work and propose new mapping and transformation rules.

12.3.2 Transformation Concepts

The meta-model approach (OMG 2003) is one of the most used transformation techniques (Fig. 12.3). It has been adapted by Bazoun et al. (2013) to the context of model transformation from BPMN 2.0 model to DEVS model. Three different levels are identified: model, meta-model, and meta-meta-model. The BPMN model is the source model to be transformed, while the DEVS model is the target model resulting from the ATL transformation. BPMN and DEVS models conform to the BPMN 2.0 and DEVS meta-models, respectively. In addition both meta-models conform to a meta-meta-model named Ecore (McNeill 2010) meta-model (developed using an Ecore-based modeling framework). A mapping is implemented by ATL between the concepts of BPMN 2.0 and DEVS.

12.3.3 Meta-Models

Source and target meta-models should be well identified to proceed with the transformation (see Fig. 12.3). BPMN 2.0 meta-model specified in OMG (2011) is the source meta-model. There is no endorsed meta-model for the target DEVS

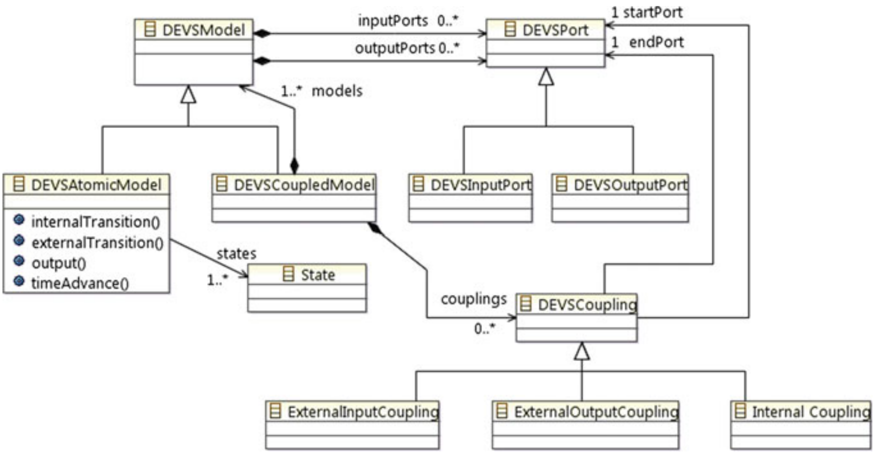


Fig. 12.6 Simplified DEVS meta-model

meta-model, but several researches were held for the purpose of building a DEVS meta-model. A synthesis work is proposed in Garredu et al. (2012). The transformation from BPMN to DEVS models has required gathering previous works for setting a DEVS meta-model, as a result the authors proposed a simplified DEVS meta-model. It is used as a target meta-model which conforms to the DEVS specification (Zeigler et al. 2000). Figure 12.6 presents the DEVS meta-model defined in Eclipse Ecore format that has been proposed in Bazoun et al. (2013).

In DEVS, there are two types of models: atomic and coupled models. Each model has a list of input and output ports. An atomic model has four main methods: internal transition, external transition, output, and time advance. A coupled model is a composition of DEVS models (atomic or coupled) and DEVS coupling. In addition, there are three types of coupling between ports: (1) external input coupling (connections between the input ports of the coupled model and its internal components), (2) external output coupling (connections between the internal components and the output ports of the coupled model), and (3) internal coupling (connections between the internal components).

12.3.4 *Mapping of Concepts*

The role of mapping in model transformation is to define links between concepts and relations from both meta-models (BPMN and DEVS). In Mittal and Risco Martin (2012), a first mapping was proposed by the authors. Nevertheless, this early mapping did not distinguish all the various types of tasks and events existing in BPMN 2.0 which differ with respect to the potential situations a task might treat.

To complete this approach, different types of task have been proposed (Receive task, Send Task, User Task, Service Task, and Manual Task); all of these tasks are mapped to "DEVS Atomic Model" concept but with different local behaviors. This is also applied to intermediate events (Receiving and Sending Messages). (Zacharewicz et al. 2008) has defined different task models. A basic task is an activity where a work is performed by a resource. For a more accurate matching between BPMN model and DEVS model it has been proposed in Bazoun et al. (2014) and then (D'Ambrogio and Zacharewicz 2016) to distinguish the "Reception Task" from the "Basic Task".

Also we clearly distinguish between tokens and messages. The structure of tokens and messages is a multi-value event as described in G-DEVS (Zacharewicz et al. 2008) that is implemented by one object with several variables. Each variable is representing one data. The notion of Event is used to represent something that "happens" during the execution of the process. It represents a step in the process and its meaning differs from DEVS event. These events affect the flow of the process. There are three types of events, based on when they affect the flow: Start Event, Intermediate Event, and End Event. In this paper we will present an example of an Intermediate Event; Intermediate Reception Event. Some information of the token will be updated by the workflow according to actions defined in the task, current values of the token, and message received. At the end, the token reflects the path taken, the duration, etc. All the data are tracked in order to compute some performance indicators. This chapter will not detail each concept, but only the most relevant are described in the following.

Table 12.1 presents a non-detailed mapping between BPMN and DEVS. It shows new concepts (*) added regarding the previous approaches in the literature introduced in Sect. 12.2.1.

This conceptual mapping has been implemented into transformation rules using ATL transformation language. Each atomic component is generated from the BPMN model than the generated components are assembled in the coupled model.

12.3.5 Tooling and Implementation

12.3.5.1 Transformation Language

An ATL M2 M (eclipse) component has been developed in the Eclipse modeling Project (EMP). The ATL Integrated Environment (IDE) provides a number of standard development tools (syntax highlighting, debugger, etc.) that aims to ease development of ATL transformations. The ATL project includes also a library of ATL transformations. ATL M2 M is also used for compliance reason with SLMToolBox (presented in the next section) both developed under Eclipse. The more exhaustive transformation rules and specifications have been introduced in a technical paper (Bazoun et al. 2014) presenting the mapping details.

Table 12.1 BPMN elements to DEVS components

BPMN	DEVS
Pool	DEVS Coupled Model
Lane	DEVS Coupled Model
Sub process	DEVS Coupled Model
Flow Message Flow* Sequence Flow*	DEVS Atomic Model
Task Basic Task Send Task* Receive Task*	DEVS Atomic Model
Event Start* {Message, Timer, Conditional} Intermediate* {Message, Signal, Conditional} End* {Message, Timer, Conditional}	DEVS Atomic Model
Gateway Exclusive Gateway Inclusive Gateway* Parallel Gateway	DEVS Atomic Model

12.3.5.2 SLMToolBox

SLMToolBox (Boye et al. 2014) is a software tool developed in the frame of MSEE project. The SLMToolBox will be used by enterprises willing to develop a new service or improve an existing one, within a single enterprise or a virtual manufacturing enterprise. The tool will be used at the stage of "requirement" and "design" of the service engineering process. The SLMToolBox is regarded to be an integration of several scientific concepts related to services into one tool. These concepts can be summarized by MDSEA methodology, services' modeling, engineering, simulation, monitoring, and control.

The simulation feature is based on model transformation from BPMN to DEVS models. Source BPMN model is extracted from the BPMN graphical editor (integrated in SLMToolBox), a transformation engine is implemented based on ATL, and the output of this engine is DEVS model. A new developed version of (Zacharewicz et al. 2008) will be integrated in the SLMToolBox for graphical visualization and simulation of DEVS models.

12.3.6 Use-Case

One use-case model from the MSEE European project has been reused to serve in this research as a case study. The process consists in the creation of a cloth patron adapted and fitted to each client by tailoring, thanks to customer data.

In MSEE project, the modeling is starting from BSM level with an Extended Actigram model. Then the next step is going down to the BPMN model at TIM level. At this level before the creation of service from the model it could be valuable to simulate its behavior in order to correct potential errors of conception that can be detected through dynamical aspects not seen by reading a static model. The next part of the section will focus on the transformation to the simulation model.

One extract from the BPMN model is detailed in top of Fig. 12.6. Two pools representing client and manufacturer are described in the use-case. In particular, the sequence and the messages exchanged with the client are considered. The distinctive contribution of this research work permits first to differentiate the type of BPMN event. For instance the model shows an intermediate "Message Event". In addition, the task 1 is emitting a message to another blind pool (with basic a reception and triggering behavior). We consider this possibility as expressing representatively BPMN 2.0 collaboration model.

At DEVS level, the LSIS_DME editor (Zacharewicz et al. 2008) was tentatively selected to perform tests on the DEVS models obtained from BPMN matching before moving to final development stage, to the DEVS engine of the SLMTooLBox. One interest for the tool comes from the fact that it enables the creation, storage library, modification, and composition of XML-based models that can be feed in our case by the ATL transformation from ATL BPMN models. Also, the editor allows editing visually a model with geometric shapes representing the different elements of a DEVS atomic or coupled DEVS model.

Mapping realized the DEVS Coupled Model based on the library developed from BPMN components (Table 12.1) and integrated in the LSIS_DME DEVS models library of BPMN diagram. The DEVS coupled model presented is the transformation results of the selected extract from BPMN model of use-case (see Fig. 12.7). Each atomic DEVS component is selected from the library and instantiated according to data values coming from the BPMN description. Then the models are coupled to represent the BPMN chain of tasks and it take into account resources represented by lanes. In this example we differentiate between a fully described lane and another non-detailed lane (blind lane).

Then Fig. 12.7 has been run to present an extract of the simulation results provided by the tool. In this simulation it was confirmed that the token variables declared in the initial state of each "start event" atomic model can be followed in term of evolution of their attributes values accordingly to activities actions of the process and regarding time. The new values depend on the operation of the task and message received. The main idea resulting from the first simulations performed is the proof of feasibility in terms of definition and monitoring of quality indicators, the capacity to measure the impact of input factors and parameters. The goal is to provide simulation feedbacks to parameters tuning to reach as close as possible the services desired results.

At the moment, results are not handled to be displayed graphically nor interpreted by BPMN. The simulation has been set up to follow performance indicators on tokens that circulate through the different components of the process. Tokens gather information on service development and its delivery; they can be considered

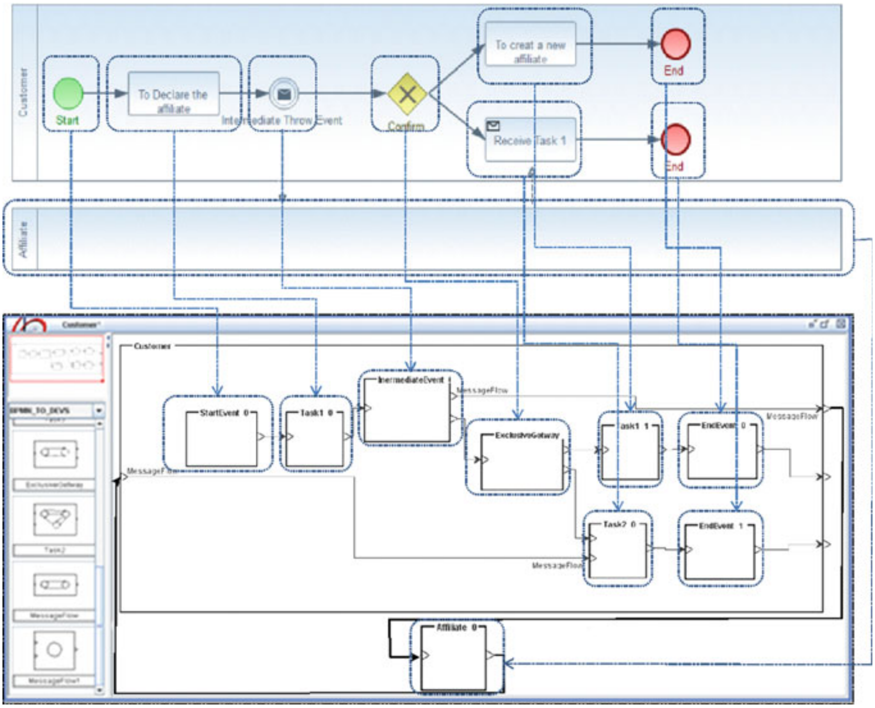


Fig. 12.7 Equivalent DEVS model example in LSIS DME

as the memory of service development. For instance, the time to complete the service delivery can be traced during the simulation. The number of resources called to achieve the service delivery process and the cost of material and human resources can be computed using the simulation. Another point is to analyze failure in the service delivery. Some service building can lead to bottlenecks. Several scenarios can be proposed and run to evaluate the best one before the next implantation step: the architecture implementation.

Nonetheless, these results are of paramount importance for decision modeling, since they provide a broader set of information (e.g., historical events, PIs and What-If scenarios) to a decision-maker.

12.4 Discussion and Perspective

Within this work, the correlation between simulation and decision-making as one of the main applications of simulation for enterprise management was discussed. Several fundamental elements, required for developing simulation-based decision-making cycle, were also presented such as GRAI decisional model.

GRAI easiness to position a decision at a specific level as well as its capability to provide a global view of the decisional perspective represent the major advantages of such an approach (Carrie and Macintosh 1997). Thanks to these characteristics, it has been applied for a large scope of different purposes (Noran 2012), such as service monitoring (Taisch et al. 2014) or analysis of business model of enterprise network (Álvares-Ribeiro et al. 2004), performance evaluation (Ducq and Vallespir 2005), and information and manufacturing system alignment (Goepp-Thiebaud and Kiefer 2008).

Some critics have been raised on this method, particularly due to the lack of dynamic modeling of information systems which does not allow modeler to show the effects of delays in decision (Carrie and Macintosh 1997). Álvares-Ribeiro et al. (2004) integrated GRAI Grid together with Zachman framework, particularly with its "Business Model" dimension. However, this conceptual integration appears limited in coping with the static nature of GRAI modeling. A more promising approach is to interconnect GRAI Grid and GRAI Nets models with simulation models. Some authors have worked toward this direction, attempting to interconnect GRAI Grid with simulation model, such as (Al-Ahmari and Ridgway 1999). Only DGRAI, an evolution of GRAI for decision systems design and monitoring, combined simulation also with GRAI Nets (Poler et al. 2002).

Inspired from this work, this chapter proposes a "*simulation-aided decision making cycle*" as an approach for coupling decisional modeling with simulation in the frame of MDSEA architecture. Within this cycle, two model transformations are to be clearly defined (see Fig. 12.8): (1) from GRAI Grid/Nets to BPMN 2.0 and (2) from BPMN 2.0 to DEVS model.

Regarding the first transformation (from GRAI Grid/Nets to BPMN 2.0), few examples of translation GRAI Grid formalism into business process modeling languages can be found in the literature. A mapping of GRAI languages for semantic translation into ULM Activity and Class diagram has been proposed (Seguer et al. 2010). There are not works aiming at translating this decisional formalism into nor BPMN neither other modeling formalism and this transformation has been addressed only conceptually in this work. A concept mapping and transformation rules are required to implement the "*simulation-aided decision making cycle*" and thus support decision-making under different perspectives. For

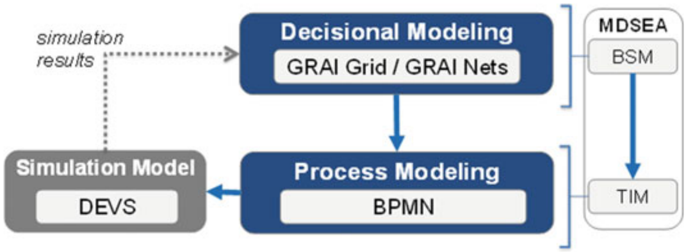


Fig. 12.8 Transformation path from decision process modeling to simulation

instance, during the running of the decisional system, simulation will help also in identifying critical activities and bottlenecks on which targeting interventions (e.g., additional supporting resources, changes in the triggers of a decisions, etc.).

GRAI Nets formalism can be also further extended with new concepts coming from decision-making theories. For instance, concepts such as decisional roles can be included based on Organizational Buying Behavior (OBB), a widespread theory used for complex decision involving groups or individuals. In practical terms, this might be done introducing a new modeling component representing a human resource, specified by an attribute “role”, which can assume values like “users”, “proposer”, “influencer”, “decider”, and “gatekeeper”. Rules and constraints in the association with the other modeling elements of GRAI Net would be then defined according to the value of this attribute

For the second model transformation (from BPMN 2.0 to DEVS) concepts, meta-models and a concrete implementation have been proposed and implemented. It remains to visualize the DEVS models resulting from the transformation to be later displayed in a DEVS Graphical editor completely integrated in the SLMToolBox. The DEVS meta-model will be completed independently from any simulator’s architecture to take into account multi-value state variables. In addition, new features such as export format will be developed. Storage will be improved. Authors claim that the durability of this work relies on the adoption of the open platform. In addition, BPMN models (subject of simulation) will be animated for better understanding of the process. Thanks to the visualization of DEVS models, users will be capable of tuning more precisely performance indicators’ values (time, costs, and combined indicators) needed for simulation. The simulation results offer sufficient information needed for business process analysis, but the problem frequently faced is the lack of temporal data from enterprises because of the domain no long experience.

12.5 Conclusion

This chapter highlighted the interest of M&S for Enterprise Management. It is intended to guide enterprise managers in the choice of appropriated M&S solutions. The chapter has also stated that models and simulations have to be closely linked; vertically, from business to technical level, and horizontally, from static to dynamic views. In these links interoperability should be ensured to preserve the information when changing or coupling different paradigms or tools. For this purpose, the interest of using architectures, such as MDSEA, based on model-driven approaches was presented. Then the chapter has focused on decisional and business process modeling and simulation in the frame of this architecture. The benefits of simulation for enterprise management and recent trends were also discussed with a focus on *simulation-aided decision making cycle*. As an example of creating simulation models in this cycle from transformation of static models, the chapter presents a transformation of BPMN models into DEVS models. It described the transformation architecture, mappings, and an implementation in an open-source tool (SLMToolBox). The development of the presented work is being followed with a special focus on the usage of the M&S results as decision aid. Eventually, the simulation of decision models is under discussion as an open perspective.

Review Questions

1. What do MDSEA, M&S, EA*, DES, BPMN, and DEVS stand for?
2. How MDSEA architecture can support resource management and development?
3. Name five advantages of simulation for enterprise management.
4. Why modeling and simulation are bundled?
5. What is simulation-aided decision-making cycle? In this cycle, what is the benefit of hierarchical decomposition of decision and aggregation of simulation results?
6. Why BPMN 2.0 models should be transformed to DEVS models before simulation? How this transformation is performed?
7. What can be the perspective of simulation as decision aid for enterprise management?

Acknowledgements This work has been partially supported by the FP7 Research Project MSEE "Manufacturing Service Ecosystem" ID 284860. http://www.msee-ip.eu/

References

Al-Ahmari, A. M. A., & Ridgway, K. (1999). An integrated modeling method to support manufacturing systems analysis and design. *Computers in Industry*, *38*(3), pp. 225–238. https://doi.org/10.1016/S0166-3615(98)00094-3

Álvares-Ribeiro, N., Martins, ÂA., & Pinto Ferreira, J. J. (2004). Introducing Time Horizons to Enterprise Networking Architecture. In *Virtual Enterprises and Collaborative Networks* (pp. 63–70). Boston: Springer, Boston, MA. https://doi.org/10.1007/1-4020-8139-1_7

Anglani A., Grieco A., Pacella M., Tolio T. (2002), Object-oriented modeling and simulation of flexible manufacturing systems: a rule-based procedure, Simulation Modeling Practice and Theory, Vol. 10, No. 3–4, pp. 209–234.

ATL (2013), ATL/User Guide—The ATL Language http://wiki.eclipse.org/ATL/ (accessed November 2013).

Bazoun, H., Zacharewicz. G., Ducq. Y., Boye, H. (2013), Transformation of Extended Actigram Star to BPMN 2.0 and Simulation Model in the frame of Model Driven Service Engineering Architecture. TMS.

Bazoun, H., Zacharewicz. G., Ducq. Y., Boye, H. (2014), SLMToolBox: An implementation of MDSEA for Servitisation and Enterprise Interoperability. In I-ESA, 7th international conference.

Benkamouna, N., ElMaraghy, W., Huyet, A.-L. & Kouiss, K., (2014). Architecture Framework for Manufacturing System Design. s.l., s.n., pp. 88–93.

Bouquet, P. et al. (2005), D2.2.1. Specification of a common framework for characterizing alignment, KWEB EU-IST-2004-507482 Project, v2.0., 2005.

Bourey, J.P., Grangel Seguer, R. and Doumeingts G., Berre, A.J. (2007) 'Report on Model Driven Interoperability', Delivrable DTG2.3, INTEROP Network of Excellence, pp. 91, http://interop-vlab.eu/deliv/tg2-model-driven-interoperability/, Avril.

Boyé, H., Bazoun, H., Belkhelladi, K. (2014), SLMToolBox: ATool Set For Service Engineering, Paper accepted in MODELSWARD 2014 2nd international conf on Model-Driven Engineering and Software Development.

Bruzzone A.G., Mosca R., Revetria R., Rapallo S., (2000), Risk analysis in harbor environments using Simulation, Safety Science, Vol. 35, pp. 75–86.

Burlton, R.T. (2001), Business Process Management: Profiting From Process. Sams publishing, Indianapolis.

Carrie, A. S., & Macintosh, R. (1997). An assessment of GRAI Grids and their use in the Strathclyde Integration Method. *Production Planning & Control*, *8*(2), pp. 106–113.

Cardoso, J., Pedrinaci, C., Leidig, T., Rupino, P., De Leenheer P. (2012), Open semantic service net-works. The international Symposium on Service Science (ISSS).

Çetinkaya, D., Verbraeck, A., Seck, M. D. (2012), Model Transformation from BPMN to DEVS in the MDD4MS Framework", TMS-DEVS, pp. 304–309.

Chen, D., Doumeingts, G. & Vernadat, F., Architectures for enterprise integration and interoperability: Computers in Industry, p. (59), pp. 647–659. 2008.

Chen, D. and Doumeingts, G., (1996), The GRAI-GIM reference model, architecture and methodology, in "Architectures for Enterprise Integration", Bernus P., Nemes L., J. Williams T. (Ed), IFIP Advances in Information and Communication Technology 1996.

Chen K. Y., Lu S. S., (1997), A petri-net and entity-relationship diagram based object-oriented design method for manufacturing systems control", International Journal of Computer Integrated Manufacturing, Vol. 10, Nos. 1–4, pp. 17–28.

Czarnecki K. and Helsen S., (2003), Classification of Model Transformation Approaches, in OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture, 2003.

D'Ambrogio, A. and Zacharewicz, G. 2016. Resource-based modeling and simulation of business processes. In Proceedings of the Summer Computer Simulation Conference (SCSC '16). Society for Computer Simulation International, San Diego, CA, USA, Article 63, 8 pages.

Doumeingts, G. (1984), La méthode GRAI [Ph.D. thesis], Bordeaux, France, University of Bordeaux 1.

Doumeingts, G., Vallespir, B., Chen, D. (1998), "Decisional modeling GRAI Grid" in International handbook on information systems, Bernus, P., Mertins, K. & Schmidt, G. ed., Berlin, Springer.

Doumeingts, G. & Ducq, Y., (2001). Enterprise Modeling techniques to improve efficiency of enterprises" in. International Journal of Production Planning and Control, 12(2), pp. 146–163.

Ducq, Y., & Vallespir, B. (2005), Performance evaluation using decisional modeling. In 4th international conference on decision (pp. 15–30).

Ducq, Y. et al. (2014), "Generic Methodology for Service Engineering based on Service Modeling and Model Transformation State of the art in model driven approaches and model transformation" in "MSEE book: Manufacturing Service Ecosystem: Achievements of the European 7th Framework Programme", FoF-ICT Project MSEE: Manufacturing SErvice Ecosystem (Grant No. 284860), Aachen: Verlag-Mainz: Bremer Schriften zur integrierten Produkt- und Prozessentwicklung, pp. 41–49.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T. & Monostori, L., (2012). Complexity in engineering design and manufacturing. CIRP Annals— Manufacturing Technology, Volume 61, p. 793–814.

Garredu, S., Vittori, E., Santucci, J-F., Bisgambiglia, P-A. (2012), A Meta-Model for DEVS Designed following Model Driven Engineering specifications, SIMULTECH, page 152–157. SciTePress.

Goepp-Thiebaud, V., & Kiefer, F. (2008). Multi-screen view and GRAI GridS to model decisional process of manufacturing IS alignment. IFAC Proceedings Volumes (IFAC-Papers Online), 17 (1 PART 1).

Hamri, M. and Zacharewicz, G. (2012), Automatic generation of object-oriented code from DEVS graphical specifications. In WSC'12. Article 409.
Jaakkola, H. & Thalheim, B., (2011). Architecture-driven modeling methodologies. s.l., IOS Press, p. 98.
Jahangirian M., Eldabi T., Naseer A., Stergioulas L. K, Young T., (2010), Simulation in manufacturing and business: A review, European Journal of Operational Research, Vol. 203, pp. 1–13.
McNeill K. (2010), "How to extend the Eclipse Ecore metamodel." http://www.ibm.com/developerworks/ library/os-eclipse-emfmetamodel/index.html
Kettinger W. J., Teng J. T. C., Guha S.,(1997), Business process change: a study of methodologies, techniques, and tools, MIS Quarterly, 1997, Vol. 21, No. 1, pp. 55–80.
Kurtev I., van den Berg K., (2005), MISTRAL: A language for model transformations in the MOF meta-modeling architecture, Lecture Notes in Computer Science 3599, pp. 139–158.
Le Moigne, J. -L. (1977), La théorie du système général. Théorie de la modélisation, University Press of France, Paris.
Lezoche, M., Panetto, H. & Aubry, A., (2011), Conceptualisation approach for cooperative information systems interoperability, in 13th International Conference on Enterprise, Information Systems, ICEIS 2011, pp. 101–110.
Lu Y. (2012), Approach for Information Systems Semantic Interoperability in Supply Chain Environment, Ph.D. Thesis, Zhejiang University, Hangzhou, Zhejiang Province, China.
Mesarovic, M. D., Masko, D. and Takahara, Y., (1970), Theory of Hierarchical Multilevel Systems, New York and London, Academic Press.
Miller J. and Mukerji, J. (June, 2003). "MDA Guide Version 1.0.1".
Mittal, S., and Risco Martin, J. L. (2012). Netcentric System of Systems Engineering with DEVS Unified Process. 610–613. CRC Press.
MSEE book. (2014). "Manufacturing Service Ecosystem, Achievements of the European 7th Framework Programme FoF-ICT Project", Stefan Wiesner, Guglielmina C., Gusmeroli S., Doumeingts G. (Ed), Aachen: Mainz: Bremer Schriften zur integrierten Produkt- und Prozessentwicklung.
MSEE D15.2, (2012) deliverable "D15.2 D15.1 Methodologies and Tools for SLM components development M9 issue".
Nethe A., Stahlmann H. D. (1999). "Survey of a general theory of process modeling", Proceedings of the International Conference on Process Modeling, Cottbus, Germany, pp. 2–16.
Noran, O. (2012). Building a support framework for enterprise integration. *Computers in Industry*, *64*. https://doi.org/10.1016/j.compind.2012.09.006
OMG, "Business Process Model and Notation (BPMN) version 2.0" doc num: formal/2011-01-03.
OMG, "MDA Guide Version 1.0." document number: omg/2003-05-01.
Pirayesh-Neghab, A., Siadat A., Tavakkoli-Moghadam R., Jolai, F. (Sep. 2011). An integrated approach for risk-assessment analysis in a manufacturing process using FMEA and DES , ICQR 2011 IEEE International Conference.
Pirayesh-Neghab, A., Etienne, A., Kleiner, M., Roucoules, L.(Sep. 2015). "Performance evaluation of collaboration in the design process: Using interoperability measurement", Computers in Industry. Vol. 72, pp. 14–26.
Poler, R., Lario, F. C., & Doumeingts, G. (2002). Dynamic modeling of Decision Systems (DMDS). Computers in Industry, 49, 175–193.
Savage, Charles M. (1996). Fifth Generation Management : Co-creating Through Virtual Enterprising, Dynamic Teaming, and Knowledge Networking Butterworth-Heinemann, p. 184.
Semini M., Fauske H., Strandhagen J. O. (2006). "Applications of discrete-event simulation to support manufacturing logistics decision-making: a survey", Proceedings of the 38th conference on Winter simulation, pp. 1946–1953,

Simon, H. A. (1969).The Science of the Artificial. MIT Press, Cambridge, Mass, 1st ed. [3rd ed. in 1996, MIT Press].

Smith J. S. (2003). "Survey on the Use of Simulation for Manufacturing System Design and Operation", Journal of Manufacturing Systems, Vol. 22, No. 2.

Seguer, R. G., Bigand, M., & Bourey, J.-P. (2010). Transformation of Decisional Models into UML: Application to GRAI Grids. International Journal of Computer Integrated Manufacturing, (september), 655–672.

Schmidt, D.C. (February 2006). "Model-Driven Engineering" (PDF). IEEE Computer. 39 (2).

Shvaiko P.(2005). A survey of schema-based matching approaches, Journal on Data Semantics IV 146–171.

Sienou A., Lamine E., Karduck A., Pingaud H. (2007) Conceptual Model of Risk: Towards a Risk Modeling Language. In: Weske M., Hacid MS., Godart C. (eds) Web Information Systems Engineering—WISE 2007 Workshops. WISE 2007. Lecture Notes in Computer Science, vol 4832. Springer, Berlin, Heidelberg.

Taisch, M., Heydari, M., Carosi, A., & Zanetti, C. (2014). Service performance monitoring and control Toolset. *Procedia CIRP*, *16*, 62–67.

Thatte, S., Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., … & Trickovic, I. (2003). Business process execution language for web services version 1.1. Microsoft and others, OASIS Standard BPELv11-May052003.

Wainer, DEVS TOOLS, hosted by G. Wainer at Carlton University, November 2013, http://www.sce.carleton.ca/faculty/wainer/standard/tools.htm

Weske, M., (2007). "Business Process Management: Concepts, Languages, Architectures". New York, Springer-Verlag: p. 368.

Wiendahl, H. & Scholtissek, P. (1994). Management and Control of Complexity in Manufacturing. CIRP Annals—Manufacturing Technology, 43(2), p. 533–540.

Wu Y-F. (2008). "Correlated sampling techniques used in Monte Carlo simulation for risk assessment", International Journal of Pressure Vessels and Piping, Vol. 85, pp. 662–669.

Zacharewicz G.; Frydman C.; Giambiasi N. (2008). "G-DEVS/HLA Environment for Distributed Simulations of Workflows", Simulation, 84(5), pp. 197–213.

Zacharewicz, G., Diallo, S., Ducq, Y., Agostinho, C., Jardim-Goncalves, R., Bazoun, H., … & Doumeingts, G. (2016). Model-based approaches for interoperability of next generation enterprise information systems: state of the art and future challenges. Information Systems and e-Business Management, 1–28.

Zeigler, B. P., Praehofer, H. and Kim, T. G. (2000). "Theory of modeling and Simulation", NY.

Author Biographies

Gregory Zacharewicz is Associate Professor at University of Bordeaux (IUT MP) with both competences in enterprise engineering and computer sciences. His research interests include Discrete Event Modeling (e.g., DEVS), Distributed Simulation, HLA, and Workflow. He has been involved in FP7 European projects (e.g., MSEE and FITMAN). He is recently focused on Enterprise Modeling and Semantic Interoperability.

Amir Pirayesh-Neghab is a research engineer in INTEROP-VLab, Belgium (operational office: Talence, France). He obtained his Ph.D. in industrial and mechanical engineering from Arts et Métiers (ENSAM), ParisTech, France in 2014. His research interests include Interoperability Evaluation, Performance Measurement, Enterprise Modeling and Process Simulation. He has been involved in a national project, called NOSCIFEL, and several FP7 and H2020 European projects such as TELLME, FITMAN, PSYMBIOSYS, and C2NET.

Marco Seregni is currently a Ph.D. candidate at the Department of Management, Economics and Industrial Engineering (DIG) of Politecnico di Milano (POLIMI) where he has been graduated in 2013. He has been participating in two EU funded projects (CTC-Close to Customer, Psymbiosys) and an Italian regional project. His main research interests include requirements engineering in Product Service Systems, supply chain management for distributed manufacturing systems, and decisional modeling for reshoring strategy.

Yves Ducq is Full Professor at University of Bordeaux from September 2008. He is working on Performance Measurement, Enterprise Modeling and Production Management since fifteen years and has published more than 20 papers in books and international journals and more than 70 papers in international conferences. He is responsible of the research group dedicated to production engineering sciences (GRAI research Group) at IMS laboratory.

Guy Doumeingts is Emeritus Professor at the University of Bordeaux, where he was appointed Full Professor and directed the GRAI/LAP (Laboratory of Automation and Productics). He chaired the Technical Committee on Computer Application in Technology at IFIP, and was member of the Editorial Board of several journals. In 2012 he was elected General Manager of INTEROP-VLab (International Virtual Laboratory for Enterprise Interoperability), a network of scientists and organizations to develop and promote Enterprise interoperability.

Part V
Learning, Education and Training

Chapter 13
Simulation-Based Learning and Education

Tuncer Ören, Charles Turnitsa, Saurabh Mittal and Saikou Y. Diallo

What is honored in a country is cultivated there.
Plato, Republic, Book VIII

Abstract Simulation is vital to many disciplines as has been shown throughout the book. Future specialists in every domain must include modeling and simulation (M&S) as integral part of their learning, education, and teaching the discipline itself. This fact has been accepted by various institutions, universities, and research centers as they incorporate M&S support to various scientific disciplines. This chapter enumerates venues that offer simulation-based education and training across broad disciplinary areas like Engineering, Natural Sciences, Social Sciences and Management, and Information Science. It emphasizes that simulation is an invaluable tool for experiential learning and teaching by performing—in silico (namely, computerized)—experiments and gaining experience.

Keywords Cognitive learning · Deep learning · Future of simulation-based education · Instructional design · Pedagogy · Simulation-based education · Simulation-based engineering education · Simulation-based information science education · Simulation-based learning · Simulation-based natural science education

---

The authors affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved for public release: Case: SM_002

T. Ören (✉)
School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Ave, 450, Stn A, K1N 6N516 Ottawa, ON, Canada
e-mail: oren.tuncer@gmail.com; oren.tuncer@sympatico.ca; oren@eecs.uottawa.ca

C. Turnitsa
Georgia Tech Research Institute, Atlanta, USA
e-mail: cturnitsa3@gatech.edu

S.Y. Diallo
Old Dominion University, Norfolk, USA
e-mail: sdiallo@odu.edu

S. Mittal
The MITRE Corporation, Virginia, USA
e-mail: smittal@mitre.org

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_13

13.1 Introduction

One of the three aims of Chap. 1 (titled: "The Evolution of Simulation and its Contribution to Many Disciplines") of this book is stated as: "To point out the fact that the phenomenal developments in many aspects of simulation (Ören 2005, 2011, Ören and Yilmaz 2012), made it an important and even a vital infrastructure for many disciplines. Indeed, time is ripe for enriching many disciplines by the transition from "model-based" paradigm to "simulation-based paradigm" to make them even more powerful. Like many other disciplines in engineering as well as in natural and social sciences, learning, education, and training are already tremendously benefiting from simulation-based approaches (Sowers et al. 1983). This transition may even be quickened and widening in scope by putting more emphasis on simulation education of not only simulationists alone but also for other future specialists in many other disciplines, including "modeling and simulation in all subjects of education, particularly teacher education" (Kazimi et al. 2013).

As outlined in Table 13.1, learning/teaching/education have many connotations. **Learning** is acquisition of new knowledge, skill, or attitude; it can be done through study, including (real life or simulated) experimentation or experience, or being taught. It is an essential ingredient of education and training and can be done in classroom, online, on-the-job, or just-in-time.

Computerized **simulation** (or computational simulation, or computer simulation, or in silico simulation, and mostly referred to, in short, as simulation) is performing goal-directed experimentation or gaining experience under controlled conditions by using dynamic models; where a dynamic model denotes a model for which behavior and/or structure is variable on a time base. So far as non-computerized simulation is concerned, as clarified in Chap. 1 of this book (Ören et al. 2017), "Simulation, in the sense of pretending (to make believe, to claim, represent, or assert falsely), has been used since a long time in relation with both experimentation and experience. Experimentation done by pure thinking is called *thought experiment* (also, conceptual experiment or *Gedankenexperiment*). Thought experiments have been used mostly in ethics, philosophy, and physics. Some examples are prisoner's dilemma and trolley problem (Brown et al. 2014). Physical aids, such as *scale models*, were also used for simulation done for experimentation purposes. Another possibility has been simulation of the real system under controlled experimental conditions, such as wheel and tire simulators." Simulation which is a vital infrastructure for many disciplines can also be very useful in learning and teaching. And has several advantages for experiential learning. In the following sections, many of these concepts are revisited within the realm of simulation-based learning. Since, the issue is learning, pedagogical principles would be beneficial to enhance several types of simulation-based education.

In this chapter, several aspects of simulation-based learning and education are clarified: In Sect. 13.2, basic concepts of learning and simulation as well as simulation-based education, in general, are highlighted. Sections 13.3, 13.4, 13.5, and 13.6 are devoted to simulation-based education for engineering, natural science,

Table 13.1 Outline of connotations of learning/teaching/education

Aspect	Different words/phrases associated with the aspect
What is learned	Information, knowledge, skill, attitude, choices, relationships
Learning	Adaptive learning, autodidacticism, blended learning, cognitive learning, collaborative learning, constructivist learning, digital learning, distance learning, e-learning, experience-based learning, experiential learning, experiment-based learning, formal learning, game-based learning, hybrid learning, in-class learning, informal learning, just-in-time learning, learning by doing, learning from experience, learning from experiments, learning from augmented-reality game, learning from mixed-reality game, learning to learn, lifelong education, lifelong learning, machine learning, online learning, on-the-job learning, open learning, personalized learning, role playing-based learning, scenario-based learning, self-learning, simulation-based just-in-time learning, simulation-based learning, simulation-based machine learning, solitary learning, student-centered learning, teacher-centered learning, technology-based learning To be informed, to be informed by an event, to be informed by an experience
Education	Pedagogy, simulation pedagogy, curriculum, lecture, literacy, illiteracy Adult education, constructivist education, constructivist education philosophy, inter-professional education, mixed-reality education, private education, professional education, self-education, simulation-based education, simulation for developing critical thinking, teacher education, vocational education
Educational goals	Affective educational goals, knowledge-based educational goals, skill-based educational goals
Teaching	Instructor, mentor, tutor; instructing, mentoring, tutoring; tutorial Student-centered teaching; self-teaching Instruction, differentiated instruction, mixed instruction, technology-mediated instruction, web-enhanced instruction
Training	Military training, technology-based training, training for health care, teachers training, virtual training, vocational training, Web-based training
Workforce development	Simulation-based workforce development (for a discipline/trade)

social science and management, and information science, respectively. Section 13.7 is reserved to discuss future of simulation-based education.

Two subjects are not covered in this chapter: simulation-based training for military as well as simulation-based training for health sciences, since two chapters (Chaps. 10 and 14) are dedicated, for these two subjects.

13.2 Simulation-Based Learning and Education

Due to richness of the field of learning and associated concepts (see Table 13.1), in this section of the chapter we point out some of the possibilities for simulation-based learning and simulation-based education. Table 13.2 lists associations/

Table 13.2 Associations/networking related with simulation-based learning, education and training

Acronym	Expanded form
ABSEL	Association for Business Simulation and Experiential Learning
CoLoS	Conceptual Learning of Science
EBEA	The Economics and Business Education Association
ETSA	European Training and Simulation Association CIC (ETSA)
IMSF	International Marine Simulator Forum
ITSA	International Training and Simulation Alliance
KTSA	Korea Training Systems Association
NICE	National Initiative for Cybersecurity Education
NASAGA	North American Simulation and Gaming Association
NM&SC	National Modeling and Simulation Coalition
NTSA	National Training Systems Association (USA)
SAGSET	The Society for the Advancement of Games and Simulations in Education and Training
SEE	The Simulation Exploration Experience
Simulation Australasia	Simulation Australasia

networking related with simulation-based learning, education, and training. A website maintained on simulation in learning, education, and training may also provide relevant information (Ören 2017a).

Even in pre-computer era, some forms of simulation have been successfully used for education and training. For example, role playing (as a type of simulation) is used for training. Similarly thought experiments provide bases for decision-making. "Tell me and I'll forget; show me and I may remember; involve me and I'll understand" says a Chinese proverb. Both real life experiments and experiences provide occasions for this type of learning by involving learners. Even though real life experiments and experiences are valuable, sometimes they may be risky, costly (including opportunity costs), not feasible, and may take a long time, in addition being haphazard. Computerized experiments and experiences provide possibilities for realistic experiments and experiences under controlled conditions.

To cover anatomy of simulation-based learning, we concentrate on five W and one H aspects of learning—namely on Who, Why, What, When, Where, and How—as outlined in Table 13.3.

13.3 Simulation-Based Engineering Education

Simulation in engineering begins with mathematical models that use physics-based methods, empirical collections, or a combination of two for balanced fidelity and complexity (Çakmakcı et al. 2017). These mathematical models can be defined at

Table 13.3 Elaborations on 5W1H aspects of (simulation-based) learning/teaching

5W1H Aspects	Elaborations
Who learns	• Beginner, advanced beginner, competent, proficient, expert, master (Denning and Flores 2016) • Student, apprentice, professional, one who needs knowledge (information) • Computer (software agent, robot)
Why learn (Goals and objectives) (Bixler and Wilson)	Types of objectives or domains of learning (Wilson): • Cognitive objectives (to increase an individual's knowledge) (Bloom et al. 1956; Anderson 2013) (being informed) (education) • Affective objectives (to change an individual's attitude, choices, and relationships) (Krathwohl and Bloom 1999) (education) • Psychomotor objectives (to build physical skill) (Harrow 1972) (training) (fine motor skills, gross motor skills; operational skills)
What to learn/teach	• Informing: Learning/teaching knowledge/information – Existing knowledge – Discovered (previously unknown yet existing) knowledge – Generated knowledge • Education: Learning/teaching attitude, choices, and relationships • Training: Learning/teaching skills – Motor skills, decision-making skills, operational skills
When to learn	• Just-in-time learning, lifelong learning • Moods that support learning: ambition, confidence, perplexity/bafflement, resolution, serenity/acceptance, trust, wonder (Denning and Flores 2016) • Moods that block learning: apathy, arrogance, boredom, confusion, distrust/skepticism, fear/anxiety, frustration, impatience, insecurity, overwhelm, resignation (Denning and Flores 2016)
Where to learn	• Classroom learning • Distance learning • Online learning • In situ learning
How to learn	• Learner learns herself from available sources (books, Web) • Machine learning system learns itself (non-supervised learning) • Teacher (instructor, tutor, mentor) teaches (informs) • Experiential learning by experiments [in vivo, in vitro, in silico (computerized)] by experience (role playing)

multiple abstraction levels to aid the learning of a relevant scientific concept. A model at a high level of abstraction is termed as lumped model. In engineering education, for specialized streams like electrical engineering, where we have

Maxwell equations, the theory is well established. Consequently, abstraction levels can be designed in an incremental manner and learning can be supported by both real and virtual systems.

Experimentation with a real system (albeit of reduced complexity in a lab setting) warrants a physical laboratory while a virtual system warrants a simulation laboratory (e.g., simulation software in a desktop setting). In this age of higher costs of university education and more accessible online education, having a simulation-based engineering education curriculum is a preferred option. Bringing both the real and virtual together for a simulation experiment is a nontrivial engineering challenge and requires expertise in hardware–software codesign and distributed simulation platform engineering (Mittal and Zeigler 2017). The primary motivation of bringing these elements together is to deliver an experience to the trainee and tutor him through scenario-based learning (Errington 2009, 2011). In the defense domain, a Live, Virtual, and Constructive (LVC) environment is used, where live assets are integrated with virtual assets with varying levels of abstraction. The virtual assets can be of identical fidelity as the real-world assets, where they are called emulators, or of reduced fidelity, where they are called simulators. An emulator adheres to the rules of the asset/system it is emulating and it behaves exactly like the real-world asset, but in a different environment. A simulator, on the other hand, behaves in a *similar* way as of a real-world asset and is implemented in a completely different way. These simulators may vary in degree of complexity and abstraction, and require model engineering. A reduced complexity simulator at a much higher level of abstraction is often called a constructive entity. Conducting an LVC event is a nontrivial exercise as there is human element present in a reasonably complex experiment. LVC environments are usually used in defense domain to bring realism to combat training in an operational context in distribute mission operations (DMO) setting (Mittal et al. 2015).

In the engineering domain, hardware-in-the-loop (HIL) environment is mostly used that corresponds to LVC in the defense domain. HIL environment usually incorporates live and virtual systems (e.g., simulators, software, and hardware) and may not consider human-in-the-loop or man-in-the-loop in the same amount of usage as in LVC training. All the established fields of engineering, (as described in Sect. 13.2 of this book) can use HIL to develop Test and Evaluation (T&E) strategies. Consequently, the path to training, education, test, and evaluation is available and customizable. However, in cross-disciplinary engineering, the path is not straightforward.

There are many emerging streams, such as cyber-physical system (CPS) engineering, netcentric complex adaptive systems (CAS) engineering, system of system (SoS) engineering for which there is not enough theory present to deliver a closed-loop solution. These disciplines are currently replete with emergent behavior as the final form of the theory is still developing. Many times, the emergent behavior is the very behavior that is desired out of such a complex system. Efficient methodologies are needed to understand the emergent behavior, harness it and thereby, make them predictable so that the training and engineering processes can be developed (Mittal 2013; Mittal and Rainey 2015). Until the theory is developed and validated, simulation-based experimentation becomes the preferred means to bring in the existing theories in relevant contexts for engineering a solution (Mittal and Martin 2017). These solutions require continuous training and feedback from existing solutions that improve the solution itself in an iterative manner.

In the era of complex system engineering, simulation-based methodologies provide a virtual environment to experiment and experience the complex phenomena and a means to investigate the usefulness of a particular solution and the solution's impact to the overall environment. The Internet of Things (IoT) phenomenon, indeed, has no existing theoretical model as the phenomenon is fairly new. How can learning and tutoring be ever attempted in engineering the new world of these super-connected ecosystems that involve human, physical systems/devices, cyber environment, and shared infrastructures such as electricity, transportation, and many others? The design of a virtual workbench is the first step to develop training, experimentation and experience in helping build the next generation of complex systems engineers.

13.4 Simulation-Based Natural Science Education

The use of modeling and simulation in education, particularly in the various disciplines under the heading of the physical sciences, comes within the slightly broader category of computational science. The US Department of Energy's Graduate Fellowship on Computational Science (in a survey of computational science and engineering education programs published by the Krell Institute) defines the term, as applied to education, as "an interdisciplinary field that applies the techniques of computer science and mathematics to solving physical, biological, and engineering problems" (Krell Institute 2016). In many cases, the phrase "computational science" is used intermittently with the phrase "modeling and simulation," especially in the context of education and research (Denning 2000). As computers become more ubiquitous in our society, it is natural that they will be given a greater use within our education systems, and within education for the physical sciences this means the use of computers to represent and solve problems. Within the scientific method this means constructing models (for different purposes), and using simulators to reinforce learning about the natural phenomena that such models represent.

It is significant to note the importance of advanced computing in scientific discovery and the place of simulation in the scientific discovery process. Advanced Scientific Computing Research (ASCR) is a program of the US Department of Energy (DOE).

> The mission of the Advanced Scientific Computing Research (ASCR) program is to discover, develop, and deploy computational and networking capabilities to analyze, model, *simulate (emphasis added)*, and predict complex phenomena important to the Department of Energy (DOE). A particular challenge of this program is fulfilling the science potential of emerging computing systems and other novel computing architectures, which will require

numerous significant modifications to today's tools and techniques to deliver on the promise of exascale science (DOE ASCR).

Another organization related with advanced computing is SciDAC (Scientific Discovery through Advanced Computing). "There are currently four SciDAC Institutes with 24 participating institutions …. The mission of these SciDAC Institutes is to provide intellectual resources in applied mathematics and computer science, expertise in algorithms and methods, and scientific software tools to advance scientific discovery through *modeling and simulation emphasis added*)." (DOE SciDAC).

However, it is also important to distinguish the following point: Classifying concepts, including disciplines, having a common aspect (in this case "computation (al),") under this common aspect may be misleading and may lead to misinterpretations and confusions. An example follows: "Computational immunology (or systems immunology) involves the development and application of bioinformatics methods, mathematical models, and statistical techniques for the study of immune system biology." (Yale immunobiology). Hence, considering "computational immunology" under the concept of "computation" would be wrong. "Computational" is an aspect of "immunology." Like mathematics which is an essential—albeit distinct—element of computation, simulation is an essential, yet distinct, element of computation.

Sixty years ago, when the first "satellite" was launched in 1957, it was called an "artificial satellite." With the advancements of the field, now the term "artificial" is not used. With the maturity of many fields, similarly, the term "computation(al)" may be dropped and the contribution of simulation may be explicit by the attribute "simulation-based."

In the process of education with the natural sciences it is typical that a student will undergo a variety of different learning activities. This is in accord with many different approaches to education, and can be represented by a variety of different interpretations of Bloom's Taxonomy of Learning Domains (Anderson 2013). It is important for students to learn about the physical phenomena of the science they are studying (chemistry, biology, geology, etc.) by studying their characteristics, constituent parameters, and the general and specific terms and definitions associated with each. But it is also important for the student to gain understanding and appreciation for how these phenomena work. This is particularly true within the context of the higher levels of Bloom's Taxonomy, where students must know enough about a phenomenon so that they can evaluate when it is occurring (and how); to be able to analyze a representation or claim about such a process; or to be able to create reports or explanations of such a process. To accomplish this, students must not only learn about the physical science, but must be able to apply the scientific method to studying it.

In both learning about a physical science, and in applying the scientific method to studying it (observations, the formation of hypotheses, and testing through experimentation), it is crucial that students can both understand, and create scientific models. This has widely become recognized as the third leg to scientific exploration, alongside empiricism (or observation of phenomena), and rationalism (the construction of theories from basic principles). A model can take two forms, that of a model of the phenomena, or a model of data about a phenomenon (Frigg and Hartmann 2017). In both cases, this can lead to a simulation, the first of which will help students to visualize what happens when the phenomena takes place, and the second will help the student understand the data that affect the process, and can lead to simulated data when results from observation are not available.

The value to education of such simulation is on several levels. First, it allows students to observe and understand a natural phenomenon when observation of the actual event is difficult or inaccessible. Some examples include biological processes that occur rarely, or at a difficult to observe scale—things happening at the molecular level, or at the neural level, for instance might be difficult to "observe" but a suitable computer simulation can assist the student in understanding what is being represented by a theory or model, and this leads to greater understanding of the phenomena itself. An example of such a simulator is the molecular workbench, the name for a collection of highly interactive molecular simulators designed to assist students with understanding difficult, but important principles about molecular dynamics, especially in biological systems (Tinker and Xie 2008). Secondly, it allows students to observe a phenomenon that is predicted by a model, but may not have occurred yet. Such an event could include a stellar process (such as the progression of stars through various sequences, leading to a possible nova or supernova), effects of perturbing an existing system that might be in equilibrium, or possible effects of activity on the environment.

An example of an education model that fits some of the above criteria is the Mitigation Simulator, available online from the Koshland Science Museum (National Academy of Sciences 2017). This is a simulator, where the student may choose several different criteria for adopting a Mitigation Strategy to avoiding bad effects of greenhouse gas emissions. The student can then explore different solutions to problems, and see how they fit within their selected strategy. In so doing, the simulator teaches the student about not only the reality of conflicting constraints on possible solutions, but also the proposed efficacy of different possible approaches to limiting or mitigating the greenhouse gases. In this way, the student gains not only insight into the effects of the gases on the environment, but also gains an appreciation for how it affects the natural world, the economic world, and the political world. Similar information could be taught using only textbook definitions and explanations (declarative cognition), but the impact of having to balance the different strategies with the selected constraint introduces the student to having to balance their choices within the model (procedural cognition).

In general, in education, but in particular in the physical sciences, the use of computational science techniques (in particular modeling and simulation) will facilitate students gaining insight into the subject matter being addressed (in this case, scientific principles and processes); students will benefit from a deeper understanding of the subject through seeing visual representations and dynamic representations of the subject matter; and students will likely become more engaged in the course (Lean et al. 2006). Such approaches have become so apparently valuable, and a part of the pedagogy, that the term for their employ is now coming to be accepted as model-based learning. Such learning includes both learning by modeling, but also requires learning to model. This, however, can be embedded into the education process by the educator, much as learning laboratory methods are germane to physical science education, learning about virtual modeling and virtual tools will gain a similar footing in model-based learning (Blumschein et al. 2009). The benefits of studying science, in silica, using virtual tools, present an enormous benefit to educational environments where resources are scarce, but in all venues, have all the benefits listed earlier.

13.5 Simulation-Based Social Science and Management Education

According to the bureau of labor statistics in the United States, Research and Development in the Social Sciences and Humanities represents 59,930 employees of which 8% are in the computer and mathematical occupations (Bureau of Labor Statistics 2017). Of these current employees, it is unclear how many have formal or informal training in the field of modeling and simulation. In the meantime, the National Center for Education Sciences reports that 531,200 bachelor's degrees were conferred in the fields of business, Social sciences, and history in 2013–2014 which together represents nearly 30% of all bachelor's degrees in institutions participating in Title IV federal financial aid program. There is a tremendous opportunity to educate the future workforce in Modeling and Simulation

In contrast, the same institution reports that the Computer, Modeling, Virtual Environment, and Simulation which is the broad category that encompasses Modeling and Simulation only graduated 291 students at all levels (certificate, two year, four year, and graduate) for the same time span. Table 13.4 shows a list of US academic institutions that award degrees in the simulation field either as a first or second major or certification. This is encouraging news because it shows an acceptance of the role of modeling and simulation in the generation and enhancement of student's abilities and marketability in the workplace.

Table 13.5 displays US research centers where students can interact and learn from the state of the art in basic and applied simulation methods within several university research centers. These centers are multidisciplinary and embody the ideal of simulation education.

However, the numbers clearly show that an approach that consists of training simulation engineers to think and investigate like social scientists and humanists is not a viable option by itself. Instead, an alternative would be to train social scientists to incorporate principles of modeling and simulation in their curriculum. Table 13.6 shows a sampling of social sciences, humanities, and multidisciplinary programs where principles of modeling are incorporated and taught.

In addition to the comprehensive inclusion of modeling and simulation in these programs, it is worth noting that other disciplines such as experimental archeology or

Table 13.4 A sampling of US institutions offering a computer modeling, virtual environment, and simulation program

State	City	Institution name
Alabama	Huntsville	University of Alabama in Huntsville
California	Los Angeles	University of Southern California
Colorado	Colorado Springs	University of Colorado, Colorado Springs
Idaho	Moscow	University of Idaho
Indiana	Hammond	Purdue University-Calumet Campus
Iowa	Davenport	Eastern Iowa Community College District
Kansas	El Dorado	Butler Community College
Michigan	Southfield	Lawrence Technological University
Minnesota	Saint cloud	Saint Cloud State University
New Hampshire	Nashua	Daniel Webster College
New York	Rochester	Rochester Institute of Technology
Pennsylvania	Moon	Robert Morris University
	Philadelphia	University of Pennsylvania
Virginia	Virginia Beach	ECPI University
Washington	Redmond	DigiPen Institute of Technology
	Seattle	Academy of Interactive Entertainment
Wisconsin	Madison	Herzing University-Madison
	Rhinelander	Nicolet Area Technical College

Table 13.5 US research centers offering opportunities in M&S education

Domain	University	R&D centers
Economics	Yale University	Cowles Foundation for Research in Economics
Engineering	Carnegie Mellon University	Center for Sensed Critical Infrastructure Research
Engineering	Massachusetts Institute of Technology	Massachusetts Institute of Technology, Engineering Systems Division
Modeling and Simulation	Old Dominion University	Virginia Modeling, Analysis and Simulation Center
Modeling and Simulation	University of Alabama in Huntsville	Center for Modeling, Simulation, and Analysis
Modeling and Simulation	University of Central Florida	Institute for Simulation and Training
Transportation	Georgia Institute of Technology	University Transportation Center
Transportation	Massachusetts Institute of Technology	Intelligent Transportation Systems
Transportation	Northwestern University	Northwestern University Transportation Center
Waste Management	Cornell University	Cornell Waste Management Institute

Table 13.6 A sample of programs teaching modeling and simulation (National Center for Education Statistics 2017)

Discipline	Description
Cognitive science	A program that focuses on the study of the mind and the nature of intelligence from the interdisciplinary perspectives of computer science, philosophy, mathematics, psychology, neuroscience, and other disciplines. Includes instruction in mathematics and logic, cognitive process modeling, dynamic systems, learning theories, brain and cognition, neural networking, programming, and applications to topics such as language acquisition, computer systems, and perception and behavior
Consumer economics	A program that focuses on the application of micro- and macroeconomic theory to consumer behavior and individual and family consumption of goods and services. Includes instruction in modeling, economic forecasting, indexing, price theory, and analysis of individual commodities and services and/or groups of related commodities and services
Demography and population studies	A program that focuses on the systematic study of population models and population phenomena, and related problems of social structure and behavior. Includes instruction in population growth, spatial distribution, mortality and fertility factors, migration, dynamic population modeling, population estimation and projection, mathematical and statistical analysis of population data, population policy studies, and applications to problems in economics and government planning
Econometrics and quantitative economics	A program that focuses on the systematic study of mathematical and statistical analysis of economic phenomena and problems. Includes instruction in economic statistics, optimization theory, cost/benefit analysis, price theory, economic modeling, and economic forecasting and evaluation. Examples: (Cost Analysis), (Economic Forecasting)
Engineering/Industrial management	A program that focuses on the application of engineering principles to the planning and operational management of industrial and manufacturing operations, and prepares individuals to plan and manage such operations. Includes instruction in accounting, engineering economy, financial management, industrial and human resources management, industrial psychology, management information systems, mathematical modeling and optimization, quality control, operations research, safety and health issues, and environmental program management
Management science	A general program that focuses on the application of statistical modeling, data warehousing, data mining, programming, forecasting, and operations research techniques to the analysis of problems of business organization and performance. Includes instruction in optimization theory and mathematical techniques, data mining, data warehousing, stochastic and dynamic modeling, operations analysis, and the design and testing of prototype systems and evaluation models. Examples: Business Intelligence, Competitive Intelligence

(continued)

Table 13.6 (continued)

Discipline	Description
Public policy analysis, general	A program that focuses on the systematic analysis of public policy issues and decision processes. Includes instruction in the role of economic and political factors in public decision-making and policy formulation, microeconomic analysis of policy issues, resource allocation and decision modeling, cost/benefit analysis, statistical methods, and applications to specific public policy topics. An example: Public Policy Analysis
Sculpture	A program that prepares individuals creatively and technically to express emotions, ideas, or inner visions by creating three-dimensional art works. Includes instruction in the analysis of form in space; round and relief concepts; sculptural composition; modern and experimental methods; different media such as clay, plaster, wood, stone, and metal; techniques such as carving, molding, welding, casting, and modeling; and personal style development

simulated dig (Brown and Fehige 2014) and simulation-based cosmology employ simulation techniques to enhance, promote, or create new skills and experiences. The future of simulation lies in its ability to connect with the large number of students in the humanities and social sciences. Already, there is a formal process taking place and informally we can say that it is accelerating. The future is bright indeed.

13.6 Simulation-Based Information Science Education

The academic topic of information science includes several closely associated disciplines including computer science, information systems, software engineering, computer engineering, and others. It also includes several topics related to information system management and the library sciences. For the education discussion, here, we will stick to the first group of topics—those related to information systems and computer science.

In studying information systems, students can be expected to learn (very broadly) (1) the skills of building (and using/maintaining) systems, (2) the use of such systems in a larger environment, and (3) the formal science and theory that are the bases for such systems. Different programs, of course, concentrate on different aspects. A computer science program might feature much more formal science and theory than an information systems program. In all three of these cases, however, the education process can profit greatly from the use of modeling and simulation.

Instructing students in the skills of understanding how systems work divides easily, in this discipline, into hardware-based systems and software-based systems. Students can be taught (establishing and reinforcing declarative knowledge) how the various components work, and can then be expected to build and use such systems (establishing procedural knowledge). In the case of hardware systems, using such tools as logic simulators, and even modeling and simulation software such as Simulink can help understand how things like circuitry and memory systems work, as well as the basics for digital system design (Yousuf et al. 2014). In the case of software systems, it is more typical to build the software itself, but even here modeling and simulation can assist the education process by providing tools such as simulated data streams, to serve as input for testing software. In learning about how systems such as database management systems work, it is typical to work with a simulated data base, usually on a smaller, more abstract scale, than a large actual database. In addition, there are simulators available for learning about languages such as SQL—a very valuable example are those language simulators available from the W3 Schools (an online resource for augmenting the education about software and markup tools valuable for making distributed systems) (W3 Schools). An overview of simulators and articles on the same can be found in (Alnoukari et al. 2013), which covers programming, architectures, digital design, and some of the subjects of the next category of instruction—especially computer networks.

The art of instructing students about how information systems work in a larger environment can focus on several different aspects. How do such systems work with each other (such as networks and distributed systems), how do such systems work with human users (such as cyber security and human systems interaction), and finally how do such systems affect their environment (such as courses of study on the impact of computers on society)? These are all questions (as a sampling) that might be answered by a course of study that can use (successfully) modeling and simulation to improve the education. In education about networks, there are many network simulators that simulate many different aspects about a network, for the student to again gain some procedural knowledge, along with the theoretical and declarative knowledge they gain from classroom study. One example in this area is the successful program Network Simulation 3 (or NS3) (ns-3). Several scholarly studies about the usefulness of using network simulation in classroom education exists, including a very good introduction to the topic found in Riley (2012), which introduces NS3 along with a competing simulator that has different features. There are simulators that involve serious gaming for a more interactive and immersive experience. These are used to teach principles of cyber security and cyber defense skills. One such example is the simulation game CyberCIEGE (Irvine et al. 2005). Some of the topics and skills that can be taught with such a simulator are briefly enumerated (Cone et al. 2006):

- Introduction to Information Assurance
- Information value
- Access control mechanisms
- Social engineering
- Password management
- Malicious software and basic safe computing
- Safeguarding data
- Physical security mechanisms.

Finally, the third grouping of education that we are looking at in this section includes the use of modeling and simulation in the instruction of formal science, and theory, especially in computer science. For this purpose, we have tools such as R and Matlab to investigate a number of subjects related to the computability of numbers, computation theory, and symbolic logic. Many of the same principles found in the section on teaching the physical sciences may apply here. An interesting addition, especially in the area of formal sciences (based on theory proposal, and refutation), is the case of the Scientific Community Game (SCG). The SCG uses the structure of a game (in the sense from game theory) to allow participants/players to serve in a game of proposing and opposing a scientific discovery, with the game rules being based on Popper's method of refutation (Abdelmeged and Lieberherr 2013). The game is developed by a group of researchers from Northeastern University. This "game" involves a model of scientific communities and how they approach problem-solving in the formal sciences. The approach could be a useful teaching tool for instruction (Abdelmeged et al. 2016).

13.7 Simulation-Based Educational/Training Activities in Other Fields and Countries Other Than USA

In Sect. 13.2, on Simulation-based Learning and Education, Table 13.2 lists associations and networking related with simulation-based learning, education, and training independent of geography. In Sects. 13.3 through 13.6, simulation-based engineering education, natural science education, social science and management education, and information science education are covered. Two related and important topics are not covered in this chapter, since they are covered in depth in two other chapters in this book. These topics are: the contribution of simulation to health care as well as health education and training, in Chap. 10 by Hannes Prescher, Allan H. Hamilton, and Jerzy Rozenblit and the role of simulation in military training, in Chap. 14 by Agostino Bruzzone and Marina Massei. In these chapters, most of the examples given are USA educational institutions. However, simulation as well as simulation-based education is practiced in many other regions and countries. In Chap. 5—Simulation-Based Cyber-Physical Systems and Internet of Things—Bo Hu Li, Lin Zhang, Tan Li, Ting Yu Lin, and Jin Cui also cover simulation education in China.

Table 13.7 displays a list of national simulation associations other than USA and China. Table 13.8 displays simulation associations by region/language.

In addition to the simulation associations listed in Tables 13.7 and 13.8, there are many research centers and military groups active in simulation (Ören 2017b).

Table 13.7 A list of national simulation associations in countries other than China and USA

Country	Society/Association
Australia	OzSAGA—Australian Simulation and Games Association
	SIAA—Simulation Industry Association of Australia
Bulgaria	Bulsim—Bulgarian Modeling and Simulation Association
Croatia	CROSSIM—Croatian Society for Simulation Modelling
France	CNRS-GdR MACS—Groupe de Recherche "Modelisation, Analyse et Conduite des Systemes dynamiques" de CNRS
	VerSim—Vers une théorie de la Simulation
Hungary	HSS—Hungarian Simulation Society
India	C-MMACS—Indian Society for Mathematical Modeling and Computer Simulation
	INDSAGA—Indian Simulation and Gaming Association
Italy	ISCS—Italian Society for Computer Simulation
	Liophant Simulation
	MIMOS (Italian Movement for Modeling and Simulation)
Japan	JASAG—Japan Association of Simulation and Gaming
	JSST—Japan Society for Simulation Technology
Korea	KSS—The Korea Society for Simulation
Latvia	LSS—Latvian Simulation Society
Netherlands	SAGANET—Simulation and Gaming Association Derneği (Medical Simulation Association)
Norway	NFA—Norsk Forening for Automatisering
Poland	PSCS—Polish Society for Computer Simulation
Romania	ROMSIM—Romanian Society for Modelling and Simulation
Singapore	SSAGSg—Society of Simulation and Gaming of Singapore
Slovenia	SLOSIM—Slovenian Society for Modelling and Simulation
Spain	AES—Spanish Simulation Society (Asociación Española de Simulación)
	CEA SMSG Spanish Modelling and Simulation Group
Sweden	MoSis—The Society for Modelling and Simulation in Sweden
Taiwan	TaiwanSG Taiwan Simulation and Gaming Association
Thailand	ThaiSim—The Thai Simulation and Gaming Association
Turkey	BinSimDer—Bina Performansı Modelleme ve Simülasyon Derneği
	MSD—Medikal Simülasyon Derneği
UK	NAMS—National Association of Medical Simulators
	UKSIM—United Kingdom Simulation Society

13.8 Future of Simulation-Based Education

Some of our views expressed in this chapter are necessarily based on trends, since they are so evident. However, based on the dictum "The best way to invent future is to invent it," we prefer to express our normative views on how disciplines can

Table 13.8 Simulation Associations by Region/Language

Region	Society/Association
Americas	CSSSA—Computational Social Science Society of the Americas
Asia	AFSG—Asian Federation for Serious Games
	ASIASIM—Federation of Asian Simulation Societies
Asia-Pacific	APSSA—Asia-Pacific Social Simulation Association
Australia /New Zeeland	MSSANZ—Modelling and Simulation Society of Australia and New Zealand Inc.
Czech and Slovak Republics	CSSS—Czech and Slovak Simulation Society
Dutch Benelux	DBSS—Dutch Benelux Simulation Society
Europe	ARGESIM (Arbeitsgemeinschaft Simulation News) Working Group Simulation News
	ESSA—The European Social Simulation Association
	EUROSIM—Federation of European Simulation Societies
	EUROSIS—The European Multidisciplineray Society for Modelling and Simulation Technology
	SESAM—Society in Europe Simulation Applied to Medicine
French	FRANCOSIM—Societe de Simulation Francophone
German	ASIM—German Simulation Society
Mediterranean and Latin America	IMCS—International Mediterranean and Latin American Council of Simulation
Pacific Asia	PAAA—Pacific Asian Association for Agent-based Approach in Social Systems Research
Scandinavia	SIMS—Scandinavian Simulation Society: DKSIM (Denmark), FinSim (Finland), NFA Norway), MoSis (Sweden)
Swiss, Austrian, and German	SAGSAGA—Swiss Austrian German Simulation And Gaming Association

benefit by being simulation-based and to realize this as soon as possible, to have simulation-based learning, teaching, training, and education to be widely adopted by them. Adoption of model-based approach by many disciplines is the right choice, since simulation itself is model-based (Ören and Zeigler 1979; Ören 1984).

As the opening quotation from Plato states "What is honored in a country is cultivated there," simulation-based activities can flourish especially in cultures (countries, institutions) that value rational decisions. In a rational World, simulation is an invaluable tool for experiential learning and teaching by performing—in silico (namely, computerized)—experiments and gaining experience. Especially in many cases, when in vivo (on real systems) or in vitro (in laboratory) experiential knowledge generation is not feasible, economical, or otherwise desirable, computerized experiments and experience (i.e., simulation) becomes very convenient and sometimes even superior to in vivo and in vitro experiential knowledge generation. Education for simulation-based disciplines may prepare future professionals to get full benefits of using simulation and can have an opportunity to enrich their disciplines.

In personal development, a good recommendation is to "Work smarter, not harder!" The proverbial sharpening the axe, as also stated by Abraham Lincoln is: "If I had six hours to chop down a tree, I'd spend the first four hours sharpening the axe." We argue that, for many disciplines, adopting the simulation-based paradigm and in preparing future professionals in these disciplines, promoting simulation-based learning, education, and training would be very beneficial. Simulation is already included in the curriculum of many disciplines. It is hoped that all conferences on education will also include some presentations/panels on the benefits of simulation in education.

To solve problems, one needs knowledge, knowledge processing knowledge, and intelligence; however, some solutions might be detrimental (to the society, even to humanity) if implemented without ethical considerations. For sustainable civilizations, an indispensable ingredient is ethics which necessitates respect to the rights of others. There are codes of ethical behavior for several professions including for simulationists (SimEthics; Ören 2000, 2002a, b). The ideal is, in professional courses (of engineering, natural and social sciences, as well as in information science), to teach relevant aspects of ethical behavior. However, in simulation-based education of several disciplines, the inclusion of SimEthics would also be highly desirable.

Review Questions

1. If you are involved in the education/training/learning of any topic, are you familiar with the benefits of simulation in this field? For example, are you familiar with several sources (such as associations, conferences, publications, software) of information about the use of simulation in education (as given, for examples at http://www.site.uottawa.ca/~oren/sim4Ed.pdf)?
2. How simulation is essential in the development of decision-making skills? Give several examples.
3. How simulation is essential in the development of motor skills? Give several examples.
4. How simulation is essential in the development of operational skills? Give several examples.
5. What are the benefits of teaching simulation concepts/techniques/software in education of future engineers?
6. What are the benefits of teaching simulation concepts/techniques/software in education of future scientists?
7. What are the benefits of teaching simulation concepts/techniques/software in education of future social scientists?
8. Why simulation is used in several aspects of health care? What would happen if simulation is not used in health-care education?

9. Why simulation is used in several aspects of military training? What would happen if simulation is not used in military training?
10. Why simulation is used in several aspects of management? What would happen if simulation is not used in management education?
11. How simulation can be beneficial in distance education?

References

Abdelmeged, A., K. Lieberherr. (2013) Crowdsourcing of Formal Scientific Knowledge. In Proc. WebSci'13, Paris, France, April 2013.

Abdelmeged, A., Xu, R., Lieberherr, K. (2016) Design and Secure Evaluation of Side Choosing Games, AAAI 2016 Workshop on Incentive and Trust in E-Communities. [Online]. Available: http://www.ccs.neu.edu/home/lieber/papers/side-choosing-games/wit-ec16/design-secure-evaluation-SCG-WIT-EC16.pdf. [Accessed 2017-02-13].

Alnoukari, M., Shafaamry, M., and Aytouni, K. (2013) Simulation for Computer Sciences Education. Communications of the Arab Computer Society, 6:1.

Anderson, L. W. et al. (2013). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Complete Edition. Pearson.

Bixler, B. Brett Bixler's e-Portfolio, PennState. Instructional Goals and Objectives. [Online]. Available: http://www.personal.psu.edu/bxb11/Objectives/GoalsAndObjectives_print.html [Accessed 2017-01-17].

Bloom, B. S. et al. (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain. NY, NY: Longmans, Green.

Blumschein, P., Hung, W., Jonassen, D., Strobel, J. (2009). *Model Based Approaches to Learning*. Rotterdam: Sense Publishers.

Brown, J. R. and Y. Fehige (2014). Thought Experiments, *The Stanford Encyclopedia of Philosophy* (Spring 2016 Edition), Edward N. Zalta (ed.) [Online]. Available: http://plato.stanford.edu/archives/spr2016/entries/thought-experiment/ [Accessed 2017-02-13].

Bureau of Labor Statistics (2017). Occupational Employment Statistics. United States Department of Labor [Online]. Available: https://www.bls.gov/oes/current/naics5_541720.htm [Accessed 2017-02-13].

Çakmakcı, M., U. Durak, G. Şendur (2017). Simulation-based Engineering, Chapter 3 of: S. Mittal, U. Durak, T. Ören (eds.), Guide to Simulation-Based Disciplines: Advancing our Computational Future, Springer.

Cone, B., Thompson, M., Irvine, C., and Nguyen, T. 2006. 'Cyber Security Training and Awareness through Game Play.' in IFIP International Federation for Information Processing, Volume 201, Security and Privacy in Dynamic Environments, eds. Fischer-Hubner, S., Rannenberg, K., Yngstrom, L., Lindskog, S., (Boston: Springer), pp. 431–436.

Denning, P. J. (2000). Computing the Profession. In *Computer Science Education of the 21st Century*. Greene, T. (ed.) New York: Springer, pp. 27–46.

Denning, P. J. and G. Flores (2016). The Profession of IT—Learn to Learn. Communications of the ACM, 59:12, 32–36.

[DOE ASCR] Advanced Scientific Computing Research. [Online]. Available: https://science.energy.gov/ascr/about/ [Accessed 2017-02-13].

[DOE SciDAC]. Scientific Discovery through Advanced Computing. [Online]. Available: http://www.scidac.gov/institutes.html [Accessed 2017-02-13].

Errington, E. P., (2009). Being There: Closing the gap between learners and contextual knowledge using near-world scenarios, *The International Journal of Learning* 16(8), 585–594.

Errington, E. P. (2011). Mission Possible: Using Near-world Scenarios to Prepare Graduates for the Professions, *International Journal of Teaching and Learning in Higher Education*, 23(1), 84–91.

Frigg, R. and Hartmann, S. (2017). Models in Science, *The Stanford Encyclopedia of Philosophy* (Spring 2017 Edition), Edward N. Zalta (ed.). [Online]. Available: https://plato.stanford.edu/archives/spr2017/entries/models-science/ [Accessed 2017-02-13].

Harrow, A. (1972). A Taxonomy of the Psychomotor Domain: A Guide for Developing Behavioral Objectives. New York: David McKay.

Irvine, C. E., Thompson, M. F.: and Allen, K., CyberCIEGE: An Extensible Tool for Information Assurance Education. Proc. 9th Colloquium for Information Systems Security Education, Atlanta, GA, June 2005, pp. 130–138.

Kazimi, A. B., A. Zehra, M. M. Sadruddin (2013). Study on the Impact of Integrating Modeling and Simulation in Teachers Training. In Proceedings of International Conference on Modeling and Simulation 2013 (ICOMZ-2013), Nov. 25–27, 2013, Air University, Islamabad, Pakistan, pp. 333–336 http://portals.au.edu.pk/icoms2013/Downloads/ConferenceProceedingsVolI.pdf.

Krathwohl, D. R. and B. S. Bloom (1999). Taxonomy of Educational Objectives, Handbook II: Affective Domain. Longman.

Krell Institute. (2016) Computational Science Education Survey. *Department of Energy Computational Science Graduate Fellowship*, Krell Institute. [Online]. Available: https://www.krellinst.org/doecsgf/cse_survey/all.php [Accessed 2017-02-13].

Lean, J., M. Moizer, C. A. Towler (2006). Active Learning in Higher Education. *Journal of Simulation and Games*, 7(3), 227–242.

Mittal, S., (2013) Netcentric Complex Adaptive Systems, in S. Mittal and J. L. R Martin, Netcentric System of Systems Engineering with DEVS Unified Process, CRC Press, Boca Raton, FL.

Mittal, S., Doyle, M., A. Portrey (2015) Human in the Loop in System of Systems (SoS) Modeling and Simulation: Applications to Live, Virtual and Constructive (LVC) Distributed Mission Operations (DMO) Training, in L. Rainey and A. Tolk (eds)., Modeling and Simulation Support for System of Systems Engineering, Wiley & Sons.

Mittal, S., Martin, J. L. R., (2017) Simulation-based Complex Adaptive Systems, in S. Mittal, U. Durak, T. Ören (eds). Guide to Simulation-based Disciplines: Advancing our Computational Future, Springer UK.

Mittal, S., Rainey, L., (2015) Harnessing Emergence: The Control and Design of Emergent Behavior in System of Systems Engineering, Summer Computer Simulation Conference, Chicago, IL.

Mittal, S., Zeigler, B. P. (2017) Theory and Practice of M&S in Cyber Environments, in A. Tolk and T. Ören, The Profession of Modeling and Simulation, Wiley & Sons (in press).

National Academy of Science, (2017) Mitigation Simulator. Washington DC: Koshland Science Museum. [Online]. Available: https://www.koshland-science-museum.org/explore-the-science/interactives/mitigation-simulator [Accessed 2017-02-13].

National Center for Education Statistics (2017). Integrated Postsecondary Education Data System. https://nces.ed.gov/ipeds.

[ns-3] Network Simulation 3. [Online]. Available: https://www.nsnam.org/ [Accessed 2017-02-13].

Ören, T. I. (1984). Model-Based Activities: A Paradigm Shift. In: Simulation and Model-Based Methodologies: An Integrative View, T. I. Ören, B. P. Zeigler, M. S. Elzas (eds.). Springer-Verlag, Heidelberg, Germany, pp. 3-40.

Ören, T. I. (2000). Responsibility, Ethics, and Simulation. Transactions of the SCS, San Diego, CA., 17:4 (Dec.), 165–170.

Ören, T. I. (2002a). Rationale for A Code of Professional Ethics for Simulationists. Proceedings of the Summer Computer Simulation Conference, pp. 428–433.

Ören, T. I. (2002b—Invited Plenary Paper). Ethics as a Basis for Sustainable Civilized Behavior for Humans and Software Agents. Acta Systemica, 2:1, 1–5. Also published in the Proceedings of the InterSymp 2002—The 14th International Conference on Systems Research, Informatics and Cybernetics of the IIAS, July 29–August 3, Baden-Baden, Germany.

Ören, T. I. (2005—Invited Keynote Article). Maturing Phase of the Modeling and Simulation Discipline. In: Proceedings of: ASC—Asian Simulation Conference 2005 (The Sixth International Conference on System Simulation and Scientific Computing (ICSC'2005), 2005 October 24–27, Beijing, P.R. China, International Academic Publishers - World Publishing Corporation, Beijing, P.R. China, pp. 72–85.

Ören, T. (2011). The Many Facets of Simulation through a Collection of about 100 Definitions. SCS M&S Magazine, 2:2 (April), pp. 82–92.

Ören, T. (2017a). Simulation in Learning, Education, and Training. [Online]. Available: http://www.site.uottawa.ca/~oren/sim4Ed.pdf [Accessed: 2017 Feb. 10].

Ören, T. (2017b). Simulation Associations, Centers, and Military Groups. [Online]. Available: http://www.site.uottawa.ca/~oren/links-MS-AG.htm [Accessed: 2017 March 17].

Ören, T., S. Mittal, U. Durak (2017). The Evolution of Simulation and its Contributions to Many Disciplines. Chapter 1 of: S. Mittal, U. Durak, T. Ören (eds.), Guide to Simulation-Based Disciplines: Advancing our Computational Future, Springer.

Ören, T., and Yilmaz, L. (2012). Synergies of simulation, agents, and systems engineering. Expert Systems with Applications, vol. 39, no. 1 (January 2012), pp. 81–88. (Available online 6 July 2011).

Ören, T. I., Zeigler, B. P. (1979). Concepts for Advanced Simulation Methodologies. Simulation, 32:3, 69–82.

Riley, G. (2012) Using Network Simulation in Classroom Education. In proceedings, 2012 Winter Simulation Conference, Berlin December 2012.

SimEthics—A Code of Professional Ethics for Simulationists, SCS (The Society for Modeling and Simulation International). [Online]. Available: http://scs.org/ethics/ [Accessed 2017-02-13].

Sowers, J. L., T. Ören, B. P. Zeigler, R. V. Houska, and R. Kopka (1983). Simulation in Education: Modeling for the Future. Proceedings of 15th Winter Simulation Conference. December, 1983, pp. 259—262. [Online]. Available: ACM Portal [Accessed 2017-02-13].

Tinker, R. F. and Xie, Q. (2008) "Applying Computation Science to Education: The Molecular Workbench Paradigm". *Computing in Science and Engineering*, 10:5, 24–27.

[W3schools]. The World's largest Web developer site. [Online]. Available http://www.w3schools.com/ [Accessed 2017-02-13].

Wilson, L. O. Three Domains of Learning—Cognitive Affective, Psychomotor. [Online]. Available: http://thesecondprinciple.com/instructional-design/threedomainsoflearning/ [Accessed 2017-01-17].

Yale Immunobiology. Computational Immunology. [Online]. Available: http://immunobiology.yale.edu/research/CompImm.aspx [Accessed 2017-02-13].

Yousuf, A., Mustafa, M., and W. Lehman. (2014) 'Electric Circuit Analysis in MATLAB and Simulink'. In proceedings, '360 Degrees of Engineering Education—121st ASEE Annual Conference', Indianapolis, IN, June 2014.

Author Biographies

Tuncer Ören is an emeritus professor of computer science at the University of Ottawa, Canada. His Ph.D. is in systems engineering from the University of Arizona. He has been active in simulation since 1965. His **research interests** include: (1) advancing modeling and simulation methodologies; (2) agent-directed simulation (full synergy of simulation and agents); (3) agents for cognitive and emotive simulations; (4) reliability, quality assurance, failure avoidance, and ethics; as well as (5) body of knowledge; and (6) multilingual terminology of modeling and simulation. Dr. Ören has over 500 **publications**, including 40 books and proceedings. He has contributed to over 500 **conferences** and seminars (about half as invited/honorary) in about 40 countries. He was inducted to SCS Modeling and Simulation Hall of Fame—Lifetime Achievement **Award**. He is also a distinguished lecturer, an SCS fellow, and AVP for ethics. He is recognized, by IBM Canada, as a pioneer of computing in Canada. A book about him was edited in 2015 by Prof. L. Yilmaz: Concepts and Methodologies for Modeling and Simulation: A Tribute to Tuncer Ören, Springer. **Other distinctions** include invitations from United Nations and plaques and certificates of appreciation from organizations including ACM, Atomic Energy of Canada, and NATO.

Charles Turnitsa is a senior research scientist for GTRI, working in the Information and Communications Laboratory. He has been involved in modeling and simulation research and education since 2004, and was involved in computational science research and development, primarily for the US Army and NASA, for the decade prior to that. In addition to serving on the editorial board for the journal Simulation, he is involved in the planning and chairing of a variety of programs and fora at national modeling and simulation conferences. He continues to be involved in teaching and researching a variety of different topics related to information systems, system's engineering, computational science, and modeling and simulation. He has worked on research efforts related to scientific modeling, data visualization, systems interoperability, and knowledge-based modeling. He has a bachelor of science degree in Computer Science, a master's degree in Electrical and Computer Engineering, and a PhD in Modeling and Simulation, from Old Dominion University. He is currently an instructor in the Professional Master's in Applied Systems Engineering for GTRI, teaches modeling and simulation (occasionally) as an adjunct assistant professor of simulation and computational science, and has been author or coauthor on a long list published research works on a variety of different topics related to modeling.

Saurabh Mittal is Lead Systems Engineer/Scientist at MITRE, founder and President of Dunip Technologies, LLC and Vice President-Memberships for Society of Computer Simulation (SCS) International. He is also currently serving as Associate Editor-in-Chief for Transactions of SCS and Editor-in-Chief of Enterprise Architecture Body of Knowledge (EABOK) Consortium. He received his M.S. and Ph.D. in Electrical and Computer Engineering from the University of Arizona. Previously, he was a scientist and architect at National Renewable Energy Laboratory, Department of Energy at Golden, Colorado, where he contributed to complex energy systems co-simulation environments. He also worked at L3 Link Simulation & Training as a contractor to US Air Force Research Lab at Wright-Patterson Air Force Base, Ohio where he integrated artificial agents and various cognitive architectures in Live, Virtual and Constructive (LVC) environments using formal Systems theoretical model-based engineering approaches. He was a Research Assistant Professor at the Department of Electrical and Computer Engineering at the University of Arizona. Dr. Mittal served as General Chair of Springsim'17 and SummerSim'15, Vice General Chair for SpringSim'16 and SummerSim'14, and Program Chair for SpringSim'15. He is a recipient of US Department of Defense (DoD) highest civilian contraction recognition: Golden Eagle award and SCS's Outstanding Service award. He has co-authored nearly 70 articles in various international conferences and journals, including a book titled "Netcentric System of Systems Engineering with DEVS Unified Process", CRC Press in 2013 that serves the areas of executable architectures, service-oriented distributed simulation, System of Systems engineering using Department of Defense Architecture Framework (DoDAF), multiplatform modeling, intelligence-based, complex, adaptive and autonomous systems, and large scale M&S integration and interoperability.

Saikou Y. Diallo has studied the concepts of interoperability of simulations and composability of models for the last ten years. He is VMASC's lead researcher in Modeling and Simulation Science where he focuses on applying Modeling and Simulation as part of multidisciplinary teams to study social phenomena, religion, and culture. He currently has a grant to conduct research into modeling religion, culture, and civilizations. He is also involved in developing cloud-based simulation engines and User Interfaces to promote the use of simulation outside of the traditional engineering fields. Dr. Diallo graduated with an M.S. in Engineering in 2006 and a Ph.D. in Modeling and Simulation in 2010 both from Old Dominion University. He is the Vice President in charge of conferences and a member of the Board of Directors for the Society for Modeling and Simulation International (SCS). Dr. Diallo has over one hundred publications in peer-reviewed conferences, journals, and books chapters.

Chapter 14
Simulation-Based Military Training

Agostino G. Bruzzone and Marina Massei

Abstract Simulation is strongly related to the Military Sector, indeed Computer Simulation has been effectively applied to Defense and Aerospace since the end of the World War II and John McLeod, founder of the Society for Computer Simulation International (SCS), applied it to Rocket Science and Aeronautics in early '50, quite a while before the advent of digital computers (McLeod in Simulation: the dynamic modeling of ideas and systems with computers. McGraw-Hill, New York, 1968). Today, after half a century, military users still represent the major stream for M&S developments. Indeed, there is a good reason why Modeling and Simulation (M&S) is so popular in Defense: this context provides usually very challenging problems with many variables and interactions, stochastic factors and heavily nonlinear systems; in addition, it is an area where competition (not necessarily commercial) pushes the innovative solutions to the edge, so all these aspects promote simulation as leading science to achieve successful results. This chapter is focused on New Challenges and Opportunities in this domain.

Keywords Intelligent agents · Interoperability · Autonomous systems · Cyber defense · Hybrid warfare · Hybrid naval training · CIMIC · PSYOPS · Strategic decision-making · CBRN · Homeland security · Humanitarian support · Logistics

A.G. Bruzzone (✉) · M. Massei
Simulation Team, Savona, Italy
e-mail: agostino.bruzzone@simulationteam.com
URL: https://www.simulationteam.com

M. Massei
e-mail: marina.massei@simulationteam.com
URL: https://www.simulationteam.com

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5_14

14.1 Introduction

Simulation as Leading Science is a very consolidated concept, as it is evident in Sun Tzu's masterpiece:

> Now the general who wins a battle makes many calculations in his temple where the battle is fought. The general who loses a battle makes but few calculations beforehand. Thus, many calculations lead to victory, and few calculations to defeat: how much more no calculation at all! It is by attention to this point that I can foresee who is likely to win or lose.

Obviously, nowadays, calculation evolved in Simulation and Computer Simulation is crucial to win battles of today and tomorrow. Therefore, Modeling and Simulation (M&S) for Defense have many areas of applications including, among others: Training & Education, Capability Development, Mission Rehearsal, Support to Operations, and procurement through Simulation-Based Acquisition. Among all these sectors, the Education and Training (E&T) is probably the most popular and common subject for applying M&S to Defense. This is true, even though Procurement could lead to even a greater impact in terms of Budget and Savings or that Capability Development could address very strategic issues (Page and Smith 1998). From this point of view, even the simulation taxonomy used in military sectors based on Live, Virtual, and Constructive (LVC) simulations is a good example of this "E&T oriented perspective." In fact, the three types used to classify the military simulation are specifically addressing training (DoD 1998):

- Live: Classical Live Simulation where real people operate real systems within a computer simulation as evolution of traditional conventional training exercises.
- Virtual: Classical Virtual Simulators where real people operate simulated systems such as sailing a virtual ship, flying a virtual plane, or driving a virtual tank.
- Constructive: Classical Computer War-gaming systems where players direct computer units.

These different types could be recombined in different ways, resulting in simulators that are, for instance, Live and Virtual, Virtual and Constructive, or even LVC. Commercially, it is quite common to present each single simulator as LVC (based on assumption that everything could be "adapted"). Therefore, it is strongly recommended to look "inside" the simulators to really understand their real nature and capabilities.

In general, the Simulation Scientists are in charge to design and define the models capable of achieving the objectives expected by final Users and Decisions Makers. They have to adopt proper levels of fidelity considering the overall problem, including different issues affecting data, computational efficiency, usability, maintainability, etc. (Loftin 1994; Fishwick 1995; Sokolowski and Banks 2011; Tolk 2012b)

Based on this approach, the Military Training uses simulation to support different kinds of E&T. The domain of learning covers different typologies such as psychomotor learning (Simpson 1972), cognitive learning (Anderson and Krathwohlr 2001), affective learning (Krathwohl et al. 1964; Goleman 1995; Philipps 2009), and social learning (Sottilare et al. 2011; Soller 2001):

- Psychomotor Learning is devoted to train the trainee to act and react properly based on physical interaction and Man–Machine Interface (MMIT) with a system. The main focus is the relationship between cognitive functions and physical ones in order to enhance physical skills such as proprioception, coordination, movement, manipulation, dexterity, grace, strength, and speed. The classical example in M&S of this typology is proposed by Flight Simulators or Tank Simulators wherein the trainee learn to become part of the weapon system and to operate it properly.
- Cognitive Learning is devoted to prepare the trainee in applying proper procedures and instructions at cognitive level. This is supposed to be effective based on the hypothesis that cognitive abilities are maintained and improved by exercising the brain, in analogy to the way physical fitness is improved by exercising the body. A classical example is to use a constructive simulation to solve a tactical problem.
- Affective Learning refers to a learning process related to students' interests, attitudes, and motivations. In simple terms, the affective learning is related to the emotional area and refers to learner beliefs, values, interests, and behaviors. So, this is concerned with trainee feelings during learning process, as well as with how learning experiences are internalized so they can guide the learner's attitudes, opinions, and behavior in the future. In this sense, Serious Games is a promising option for military applications (Michael and Chen 2005).
- Social Learning is a learning process within the social context. It is supported by observation and training and reinforced by the impact of social roles and networks in the individual learning. The social learning is strongly affected by fundamental skills required to military Commanders and it benefits strongly from collective training and interoperable simulation allowing to be immersed into a large exercise where these conditions are present as it happens during a Computer Assisted Exercise (CAX) or Simulation Multi Coalition Simulations (Bruzzone et al. 2015d).

Learning processes lead the E&T in military sector based on different modes such as the following ones (Cayirci and Marincic 2009a; Sottilare 2009; Tolk et al. 2012a; Stevens and Eifert 2014):

- Individual Training: This is devoted to prepare the single trainee to react and act within a scenario in a stand-alone mode. Therefore, in this case also, it is possible to have Computer Generated Forces (CGF) controlled by computer Artificial Intelligence (AI) devoted to reproduce opponents or friends, but training is focused on the single individual trainee.
- Collective Training: This is devoted to prepare a team of warfighters with same or different roles, to act and react within a scenario together emphasizing the

importance of self-coordinating and of creating harmony and synchronization in order to build up the whole team.

- Tactical Training: This is devoted to prepare Officers in dealing with tactical scenarios, usually operating on constructive simulators with high resolution.
- Theater Level Simulation: This is devoted to prepare the Headquarters in dealing with Operational and Theater Level Scenarios, usually operating on constructive simulators with high aggregation level (e.g., minimum units corresponding to brigades or battalions).
- Strategic Training: This is devoted to E&T of Commanders on the Strategic Decision-Making Processes.

These types of training could be combined in different ways, resulting in very different systems: from an individual Portable Anti Aircraft Simulator to an Air Defense War Room Training Equipment for Collective Training using Distributed Interactive Simulation (DIS), from a Virtual Bridge Simulator to a Joint Naval Training based on LVC using High Level Architecture (HLA) simulation enabling interoperability among real assets (e.g., c2 systems, vessels, and aircrafts) and simulation models (e.g., foe, friendly and neutral CGFs, hostile missiles, and generic traffic).

A very common case of Military Training is represented by CAX that involves a large number of people including trainees, operators, instructors, etc. (Cayirci 2009b). CAX is very important to train a Command to operate in scenarios by organizing exercises that are realistic. Currently, the main focus in these cases is on procedural aspects. Therefore, often multiple elements are combined together resulting in very complex mission environments that requires long time to be prepared and efforts to be conducted. In these cases, the After Action Review (AAR) is a critical issue and simulation has a big potential to improve the E&T processes by introducing smart reports and review of the actions. There are opportunities for advancement in this area by improving the impact on the training audience by new technologies. In fact, it is important to outline that M&S is a revolutionary approach in Military Exercises for over 25 years drastically reducing the cost of training by extending the use of Simulation: thanks to the introduction of distributed and interoperable simulation (Thorpe et al. 1987; Pimental and Blau 1994; Miller and Thorpe 1995). Traditional military exercises represent examples of "manual simulation" (e.g., driven just by human judges and observers) as they happen with the Role Play Game (RPG). Vice versa, the innovative aspect is the concept to introduce Computer Simulation (even if we just refer to it as to Simulation) and to use models implemented in software applications dealing with calculation of system capabilities, actions, and reactions (Nam 1980; Graetz 1981; Malley 1984). The original concept was introduced in the beginning of last century by H.G. Wells by defining the rules and probabilities regulating war games while playing with friends using the toy soldiers of his sons (Wells 1911, 1913). The modern algorithms allow introduction of much more detailed and sophisticated computations, covering, for instance, friction and logistics needs of a unit or of a weapon system (Lanchester 1995; Gozel 2000).

By this approach, the Computer Simulation allows creation of very large exercises with just a limited number of real units and assets, reducing operational costs, preparation costs, improving safety by operating on the virtual context (Orlansky et al. 1994; Rolands et al. 1998). These aspects have further been reinforced by developing simulation standards able to support distributed simulation (e.g., DIS at beginning of '90) and interoperability (e.g., HLA since 1996), allowing the development of new training simulation solutions (Hamilton et al. 1996; Strassburger et al. 1998; Zeigler et al. 1999; Huan et al. 2003; Mittal et al. 2008). In this framework, HLA evolved along the years becoming the edge technology for simulation interoperability in use for training and has been constantly updated through the decades. Even after 20 years from its inception, HLA still represents the reference standard: thanks to its open architecture approach, despite the ongoing research for developing new approaches (Fullford 1996; McGlynn 1996; Ratzenberger 1996; Möller and Dahlin 2006; Gustavsson et al. 2009; Martínez-Salio et al. 2012).

In the last two decades, the simulation standards, with special attention to the critical aspects of interoperability issues, improved capability to combine real equipment with different kind of simulators. Distributing them over the web resulted in a drastic improvement in realism and reduction of costs (Gibson et al. 2003; Liu et al. 2016).

Therefore, it is important to outline that the complexity of the scenarios and models keep this kind of interoperable distributed simulation in the area of challenging simulation developments, even today. This is not just due to the technological challenges of distributed interoperability. There is a strong need to educate new generations of scientists, engineers, developers, and technicians. In this sense, there are very important active initiatives at international levels such as Simulation Exploration Experience (SEE), formerly known as SMACKDOWN, promoted by NASA in joint cooperation with Academia and Industries to bring together the HLA standard and various other technological advances (Elfrey et al. 2011; Bruzzone et al. 2016f).

Probably, the most crucial aspect is related to the complexity of the models and the conceptual interoperability that introduce critical integration problems, if the conceptual models have been developed improperly (Lorenz 1993; Amico et al. 2000; Tolk and Muguira 2003). In several cases, this is motivated directly by the original complexity of the real systems. As an example, a main battle tank (MBT) needs gasoline to properly operate (logistics support), to properly receive its orders (command and communications) and to have different kind of ammunitions and consumables (cannon, guns, and grenade launchers); it should consider its operational status (e.g., its integrity); it has to take care of boundary conditions with respect to its equipment and situation (e.g., specific terrain type versus track conditions, sensor performance versus weather conditions), opponent countermeasures, fog of war, etc. In addition, to these elements, each good MBT Commander will outline the importance of crew and human factors for the success of the mission. All these considerations provide a good example of the dimension of the problem, the number of variables and the scope of different kind of elements to model even for a basic case. Indeed, just considering the abovementioned issues, corresponding to the tactical elements characterizing a tank battalion, it is evident that to prepare a theater level scenario on an already consolidated simulation system, it could take many weeks using dozen of experts. In a similar way, many CAX operators could be required to execute the simulator during an exercise with a real Command.

In fact, the workload to prepare and execute exercises is still an open issue for simulation and new technologies are expected to provide improvements in future. There is a clear need to reduce time, personnel, and efforts for preparing and managing exercises as well as to create models that are able to cover new areas. For instance, the use of Intelligent Agents could allow automating the control of several units, being able to assigning them high level tasks and letting them to autonomously proceed in carrying out low level orders (Bruzzone and Massei 2007a).

In a similar way, the use of Serious Games (SG) allows improved training effectiveness and the achievement of training objectives by adopting strategies enhancing trainee engagement (Abt 2002; Iuppa and Borst 2006). It could be useful to consider an example to better understand the importance of engagement: an officer required to participate in a CAX, sometimes considers it quite boring and experiences limited engagement during it and along the AAR. Vice versa, a young generation gamer may spend many hours in playing for its own gratification, often sacrificing his free time to it. This makes it clear that the engagement mechanism has the potential to maximize the effectiveness of training and to create additional training opportunities (Raybourn 2009; Raybourn et al. 2010; Bruzzone et al. 2009b).

Along the last decade, the NATO M&S COE in cooperation with Allied Command for Transformation, organized the NATO CAX Forum to support CAX Community. This event resulted in evaluation of the most important frameworks and proposed advances in this specific area, often combined with other scientific events such as the International Workshop on Applied Modeling and Simulation (WAMS) in order to address the technical evolution in the sector (Bruzzone et al. 2012a).

Even thinking of other major Simulation Events, such as Exhibitions that are going around the world, it is common to witness the maximum attendance where the E&T are specifically addressed, as happens for the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) and the International Training Equipment Conference (ITEC). In these events, several thousand visitors are registered each year, even though many other scientific events consider these aspects (e.g., I3M International Multidisciplinary Modelling and Simulation Multi-conference, Summer Computer Simulation Conference, International Defense and Homeland Security Simulation Workshop, SpringSim, WinterSim, etc.). Indeed, there is obviously a clear motivation for consolidation of this mindset about military training based on simulation and it probably is laid well by Confucius's:

> Tell me and I forget. Teach me and I may remember. Involve me and I will understand (Lau 1979).

Military needs to be trained to properly fight ("train as you fight") as well as to operate modern weapon systems. For sure, war is not a framework where training on the job could be adopted effectively, considering the very high death rate of untrained people in the battlefield. Due to this reason, it is evident that M&S represents the most effective way to prepare the warfighters and that simulated scenarios are much more realistic than traditional exercises. Indeed, in modern Warfare, simulation is often able to reproduce the real operations with high fidelity.

The importance of Military Training is becoming more and more crucial along recent years for several reasons that are hereafter summarized:

- Costs are probably the major driver in extending the use of computer simulation for Military Training considering the flight costs for a modern plane (e.g., 68,000$/h for a F-22 almost three times more than a F-16), but even for a Drone (4700$/h for a Reaper Drone) without mentioning the prohibitive costs of a ship (easily several hundred thousand per day for a vessel and over 1 million/day for a conventional carrier) or of an army division deployed on exercise (Lundquist 2009; Thompson 2013).
- Training Effectiveness is strongly improved by introducing distributed interoperable simulation: fighter pilots could train in joint operations against opponent wings, soldiers could fight against tanks and artillery in scenario overpassing the boundary of training areas, ships could carry out missions as part of a task force within intensive scenarios. These kinds of operations are often not possible in reality due to limitations imposed by available space, available resources, and safety regulations. The paradox is that, sometimes, the simulation is more "operationally realistic" than the traditional exercise.
- New War paradigms are emerging that require training people at all levels on how to deal with these new contexts. For instance, warfighters had to focus mostly on urban operations and interactions with civilian (e.g., CIMIC). These are new operations not in the traditional core of military training (Zaalberg 2006).
- Equipment and Weapon Systems are becoming more and more sophisticated and require specific training and knowledge for being properly operated. This is easily confirmed by examples such as the use of a man-portable air defense system or the capability to fly a micro drone within an urban environment.
- The Warfare has become very "invasive," also due to the evolution in weapon lethality (IED) and different gears (e.g., protective plates, communications, CBRN). Untrained people are vulnerable and could generate high rate of casualties and mutilations among rookies, almost unacceptable in most evolved Countries.

It is evident that the continuous increase of weapon systems, as well as their maintenance, is opposing to a continuous reduction in military expenditures that could be sustainable just by adopting highly efficient training at low costs, to leave resources available for real operations that unfortunately, are also incumbent in modern times.

That is to say, the roman adage “si vis pacem, para bellum” (“If you want peace, prepare for war”) is the big driver for having extensive simulation for military training.

Therefore, the military sector is currently subjected to a general change, or transformation, by breakthroughs in innovative concepts. The evolution of autonomous systems and new domains (e.g., cyberspace) represent a revolution that will require Training Simulation to maintain a key role for preparing the new generation to deal with these new contexts and new models. Synthetic environments will have to be developed in order to address these issues. From this point of view, it should be noted that the concept originally used as motto by the STRICOM (now named PEO-STRI): “all but war is simulation” is still valid. This sentence means that simulation could be extended to cover all new areas, even if, obviously, on the real war there are specific elements that are not possible to properly reproduce in computer simulation such as massive and individual human factors, spirit of survival, extreme details, etc. (Bruzzone et al. 2007b).

Simulation always requires approximations in order to be effective, reliable, and usable in accordance with its specific objectives. This fundamental aspect is acceptable in defense and does not limit the validity to use M&S for Education and Training.

In reference to this aspect, the short example of a strategic bomber simulator developed several decades ago for training the crew should provide a good illustration. In this case, the focus was on Electronic Warfare (EW). Therefore, the penetration among the enemy territory was short and intensive, considering that at that time the scenario was focused on nuclear war and the use of first generation of cruise missiles ($\sim$2000 km range). To guarantee commitment of the training audience, the full crew was simulating the whole mission: from departure from homeland, carrying out air refueling spending several hours before to enter into enemy territory, making them aware that a mistake will cost a lot of time to restart it (Thorin et al. 1982). In the following examples, the use of Simulation for Military Training is presented in relation to many specific areas, to present the potential and effectiveness of M&S in this context and current advances. All the material presented is in public domain. Therefore, even these simple examples confirm the strategic advantage provided by M&S.

14.2 Saturation Attack Against a Task Force: Joint Naval Training, VV&A, and Simuland

One of the most common scenarios in naval simulation is a ship, or task force, facing a saturation attack. A sufficient high number of threats (e.g., planes, missiles) could saturate the defenses of a group of warships resulting in breaking through and damaging/destroying the vessels (Bradford 1992; Townsend 1999; Boinepalli and

Brown 2010). Originally, this kind of analysis was carried out by static models, or simple computation considering the maximum number of targets that the different fire direction could track along with the characteristics of the weapon systems. This computation is approximate considering the dynamics of the case, especially, when the scenario moves from a classical single vessel to a naval task force and additional details are considered. For instance, cooperative engagement could improve defense capabilities if the action is coordinated among defenders. Similar techniques are also supposed to evolve in new generation missile and autonomous systems threats (Yang et al. 2013). Today a scenario like this one strongly relies on joint cooperation among vessels and other assets that are able to provide all-weather tactical warning services against such threats. ICT models, in this case, are very critical, considering the complexity, speed, and precision required to deal with this kind of scenario.

The necessity to use simulation to reproduce this context is quite evident. At the same time, such simulation represents a very valuable asset to train crew and console operators to face such challenges. It is also evident that the opportunity to adopt distributed and interoperable simulation enables training multi-ship cooperative engagement. This approach enables the possibility to have people operating from air defense console simulators interconnected within real ship operations rooms. This example outlines the advantage to move from a single stand-alone simulator of a tactical console to train multiple operators on an air defense scenario in order to be able to play much more realistic cases.

To better understand this case, consider the following example: a small task force composed of six surface vessels (i.e., a medium size aircraft carrier, two destroyers, two frigates, and an auxiliary ship), plus a diesel submarine and several autonomous systems including Autonomous Underwater Vehicles (AUV), Unmanned Surface Vehicles (USV), and Unmanned Aerial Vehicles (UAV), proceed within a dangerous area with coverage provided by aerial traditional and/or innovative solutions (e.g., E-2C for Airborne Early Warning and Control or AEW drones). This task force has to face an enemy joint attack combining cyberspace and naval joint layers. The physical threat is composed of a swarm of new generation supersonic anti-ship missiles fired by a SSGN (Nuclear Guided Missile Submarine) sailing 200 nautical miles away. The new generation missile is supposed to have supersonic speed, sea skimmer profile, high maneuvering capability as well as an intelligent control system enabling them to communicate during the attack in order to share targets and redefine priorities. The simulation of this scenario has to cover very different cases: from an intense saturation attack (24–32 missiles), corresponding to cold war cases, to a hybrid warfare context where the attacker pretend to have fired (few, or even, a single missile) just due to an error. These cases are combined with synchronous and asynchronous cyber attacks capable of effecting and even compromising the Airborne Early Warning (AEW) systems, Command and Control (C2), SATCOM & Communications, and Command Chain (Bruzzone 2016e). In case the AEW fails, the reaction time for the task force could be reduced to 10–15 s, breaking down the antimissile defense capabilities of the group, so the ship defense, adopting cooperative engagement, turns to be very stressful in terms of data communication and fire direction. Therefore, in this situation also, the supposed offensive cooperative engagement system of the missiles will have to deal with very challenging conditions. To be able to provide additional capability to the attackers (in the system) with respect to traditional target tracking, they have to coordinate multiple missiles flying, as an attack pack, at 1000 m/s, 5–10 m from sea surface within an intensive EW environment. These conditions provide limited time to get the whole picture and to renegotiate targets among the missiles based on the evolution of the scenario, i.e., to reassign the aircraft carrier as primary target for two missiles as soon as the preassigned ones miss it or are shut down.

These elements are very challenging in reference to mere physical considerations such as admissible accelerations and dynamics, but become even more complex and uncertain in reference to variables such as the effectiveness of missiles to acquire targets or to resist to EW. Finally, the missile intelligent control, as well as their capabilities to communicate and cooperate, are approximate and their performance is subjected to Subject Matter Expert (SME) estimations characterized by large confidence bands. It is evident that this scenario deals with many complex factors, including a lot of data that should be just estimated. Let us consider the traditional ones: the characteristics of this new generation missile are unknown or just estimated by intelligence reports and expert analysis. The defensive heterogeneous network among AEW, UxV, vessels of the task force, the ICT networks on board of the ships plus the whole sensor/communication network in charge of data collection, elaboration, and decision-making (from ships to ground HQs of the different Services and Authorities) is even more complex and represent other elements that need to be simulated as they are sources of potential vulnerabilities.

This case is presented in reference to the need of properly validating simulation models. In fact, in reference to missile and antimissile simulation, traditionally it is stated that the hardest part of a simulation project lays in the validation. Sometimes, it requires including real Hardware- and Software-in-the-loop to acquire useful information for this purpose (Jackson et al. 1997). In some cases, it could be even necessary to conduct a real test by firing real missiles to collect data for validating the model, corresponding to a cost in USD millions. All these impressive efforts should be evaluated in terms of fidelity and sustainability, without forgetting that firing a missile within a test could not be a perfect representative of a real scenario especially considering factors such as reliability and performance of all the systems involved and potential real boundary conditions. Despite these challenging considerations, the presented case is even more complex considering that it is not possible to adopt HIL (Hardware in the Loop)/SIL (Software in the Loop) due to the fact that most of the innovative systems to be simulated are Secret and/or belong to a Potential Opponent Nation. We cannot ask them to fire the missiles while we measure all data, at least, until they are not at war. So, let us consider how Simulation could be used to address this problem and what are the fundamental principles and the crucial elements to be considered. As anticipated, among the most critical parts of simulation development processes, Verification Validation and Accreditation (VV&A) emerges as a key element and strongly reinforced within Military Training.

VV&A relies on strong cooperation among experts of very different domains and requires that simulationists acquire deep knowledge of their context and capability to interact with other fields (Szczerbicka et al. 2000; Sarjoughian and Zeigler 2001a). Indeed, in this area, it is very critical to avoid the "negative training" (Amico et al. 2000). The "negative training" corresponds to the case when the training equipment provides an improper representation that allows the trainee to succeed in his mission by applying solutions, behaviors, actions that do not work in reality and vice versa, where proper reactions in reality could lead to failure in the Simulation (Page and Smith 1998). So, in the hereafter presented, saturation attack case, improperly setting the characteristics of the threats could lead to completely incorrect training procedures and experimental results. In addition, the negative training could be combined also with other kinds of défaillances (e.g., failing in simulation objectives, inadequate synthetic environment, synchronization errors, approximations) that need to apply proper VV&A methodologies to be prevented (Balci 2004). VV&A is based on consolidated methodologies and procedures and is a fundamental part of M&S development (Balci et al. 1996; DoD 5000.61; Youngblood et al. 2010). From this point of view, it is crucial to consider that VV&A should be adopted across the life cycle of the simulation development in order to achieve:

- Validation: process to assess the degree to which the adopted Models, Simulators, and related Data provide a proper and accurate representation of the Real World with respect to the specific intended uses of the Simulation.
- Verification: the process to assess if the Models, Simulators, and Databases are consistent with the conceptual descriptions and specifications.
- Accreditation: the process that guarantee the acceptance of the Models, Simulators, and related Data by the User for his own needs, as well; when applicable, the formal certification that the simulation is acceptable for the specific intended use.

These concepts are strongly related to the necessity of creating proper conceptual models (Validation), implementing that correctly (Verification) as well as to guarantee that final users trust the Simulator and use it (Accreditation). Conceptual model definition as well data collection is very critical (Amico et al. 2000; Zacharewicz et al. 2008). It is very important to stress that, in general, and based on very valuable experience, the major cause for simulation failure is exactly on these issues and more specifically, in losing the trust on the models of the decision makers and users (Williams 1996). It is evident that, people, in order to use simulation, need confidence that could be achieved just by properly involving them in the simulation development process since the beginning: from objective definition till the check of the experimentation results. This aspect strongly relies on the capacity of Simulation Experts to interact and to apply VV&A techniques to share information that creates this mutual model trustworthiness without getting lost in too much technical details (Amico et al. 2000).

It is even important to mention the concept of Simuland, introduced by John Mcleod almost half a Century ago, but still very actual. To validate and verify a model or simulator, it is required to consider the Real System as a reference and even when this system does not exist yet (e.g., a training simulator for air defense against a potential new missile), the experts have to assume hypotheses or to use numerical models to face this issue.

However, the point is that it is impossible to know what the real situation is. Vice versa, the situation is much more common when the Subject Matter Experts (SME) do not have an exact knowledge of the real system, but just partial information. So this approximated knowledge of the reality is defined as Simuland and Simuland is adopted to conduct the VV&A and to develop the simulation. This means that during dynamic experimentation, and along the final phases of Simulation developments, it becomes even more important to conduct tests to verify that the Simuland was correctly estimated with respect to the Real System at least in reference to the specific simulation objectives. Figure 14.1 illustrates the relationship between various elements.

The earlier presented case fits exactly in this context: a simulation dealing with defense against an opponent weapon system that is not very well known. In this case, it is required to adopt and test different assumptions based on hypotheses and intelligence reports as well as to evaluate alternative new systems, still in development phases, as countermeasures.

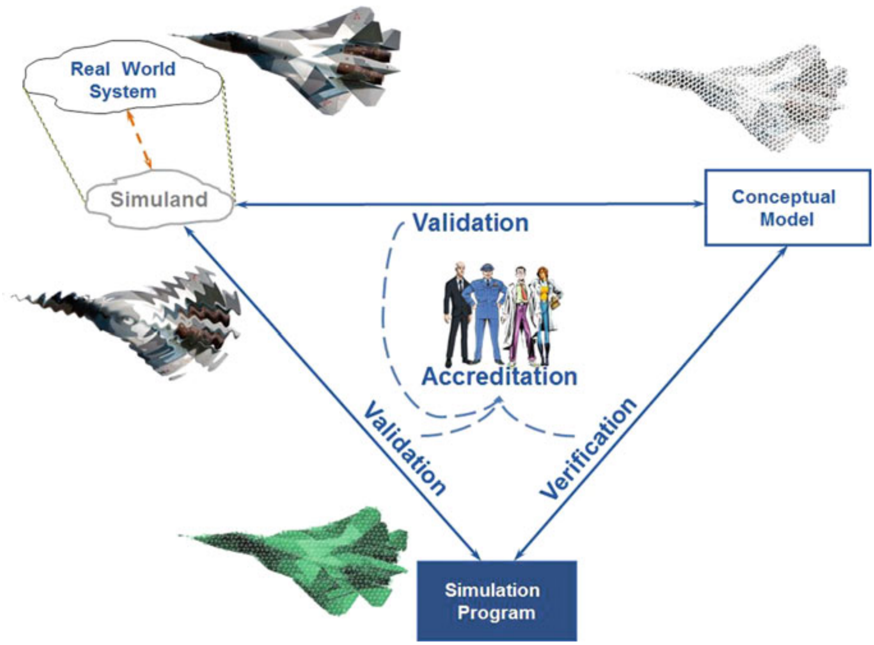


Fig. 14.1 VV&A processes along Simulation Life Cycle

The use of autonomous systems in the defense architecture represents an innovative aspect and proper modeling of these components should be done. In this case study, the simulator, Joint Environment for Serious Games, Simulation and Interoperability (JESSI), was used as virtual environment with the aim to simulate and understand Joint Naval Scenario over the Extended Maritime Framework (EMF) including sea surface, underwater, air, land and coast, space, and cyberspace (Bruzzone 2016e). The concept of EMF has been developed in specific reference to advances of M&S in Maritime Domain in Academia and International Organizations (Bruzzone et al. 2004, 2013b; Milano 2014). In this scenario, not only the systems and behaviors, but also the environmental conditions are very challenging to model. There are multiple elements that are spread over the map and affected by complex dynamics such as sea conditions, wind, waves, fog, rain, salinity and thermal layers, etc.

Due to the complexity of the system and its extensions, scalability is a very crucial element. The concept of "fog of war" introduce by Von Clauswitz "Vom Kriege" (1832) in English language "On war" represents a challenge in the estimating and modeling of opponent tactics and equipment, but also is necessary to create a simulator that is able to address multiple levels of resolution moving from operational procedures, policies and tactics down to capabilities of new kind of UxV (Unmanned various domain Vehicle) and the associated Hardware and Software in the loop issues (Bruzzone et al. 2013e). Special attention should be given to the integration of Autonomous Systems with traditional assets. For instance, just considering the problem to model the communications among the entities; JESSI simulates them through a dynamic network where nodes and links are interconnecting real and virtual assets considering their conditions and status. Each of the nodes and links composing the networks is defined in terms of type (e.g., Radio frequency, SATCOM, Acoustic Modem, optic fiber, etc.). A basic model incorporating bandwidth, capability, standard background traffic model, reliability, availability, confidentiality, integrity, mutual interference is built. All data exchanges are simulated as packets, routed over different heterogeneous networks based on the evolving situation. JESSI allows visualizing both of them on the real layer and in the cyberspace as proposed in Fig. 14.2. The status of a node or a link could be visualized for SME in order to provide an augmented reality feature, which improves their understanding of the complex situation of communications. By observing the dynamic movements of the flying cubes representing the data packet, it is possible to immediately understand the evolving situation of the heterogeneous network that is affected by changes of high order of magnitude in communication speed (e.g., from Link-16 down to 150 bauds of underwater communications in some conditions) and breakdowns. In addition, the communication model considering the network performance is not only based on the dynamic configuration of the network (e.g., AUV on surface using Radio Frequency versus acoustic modem when submerged), but also incorporates the general traffic on their nodes and link. For instance, standard traffic models could be used to reproduce the stochastic bandwidth saturation due to additional communication packages transferred during operations and not covered by current simulation. This is an example where simplified metamodels for

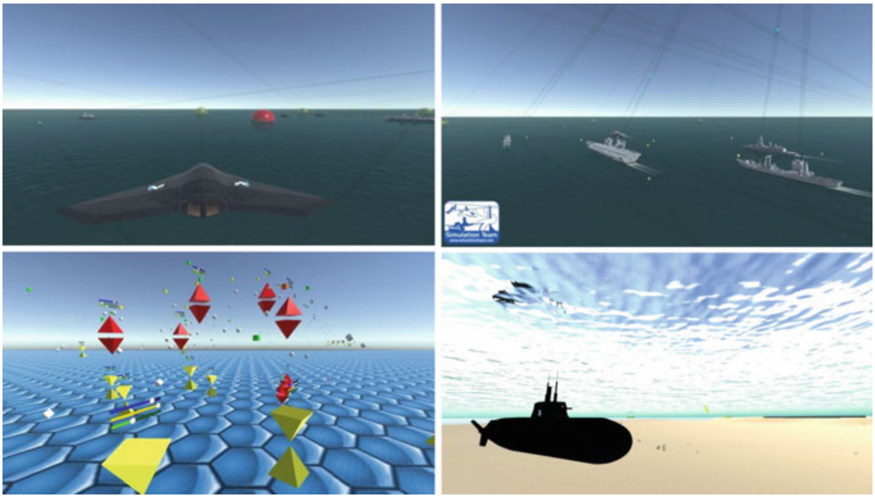


Fig. 14.2 JESSI presenting scenario evolution over different domains

communication could be effective for creating realistic test cases while more sophisticated ones could be federated in JESSI Federation to guarantee a high fidelity simulation for detailed large communication architectures (Mak et al. 2010; Bruzzone et al. 2015a).

JESSI adopts the innovative paradigm of Modeling, interoperable Simulation, and Serious Games (MS2G) that combines M&S fidelity, simulation interoperability, and intuitive and engaging characteristics of SG and was introduced by Simulation Team & Sim4Future (Bruzzone et al. 2014b). Indeed, in MS2G paradigm, M&S and SG are combined by integrating different models and to create virtual worlds easily deployable with multiple distributed solutions through HLA (Bruzzone et al. 2014c). JESSI was originally designed to simulate complex scenarios in defense and homeland security and over multiple domains where it is necessary to provide SMEs with an intuitive immersive environment to investigate the context. This approach enables application of reverse engineering based on M&S to influence the system modeling, working side by side with experts. In order to check consistency among different hypotheses about system performance and alternative behaviors, the simulation dynamically calculates the Measure of Merits (MoM), including overall mission effectiveness. The outcome is dependent of the different variables and dynamic behaviors present in the scenario.

In our case, the goal is to conduct virtual experimentations to evaluate consistency of different hypotheses on friend and foe systems as well as on Concepts of Operations (CONOPS), strategies, policies, and technological alternatives. By this approach, it becomes possible to improve understanding of this Mission Environment and to create a flexible training equipment for Crew and Commanders.

To succeed, it is important to consider the capability of the simulator to create an interoperable synthetic environment integrating hybrid stochastic simulators through HLA. JESSI evolved from Simulation Team Virtual Marine (ST_VM), previously developed for marine operations (Bruzzone et al. 2011a). The use of HLA is motivated by the need to keep an open the approach by adopting the up-to-date standard in interoperable simulation and to guarantee flexibility (Kuhl et al. 1999; Joshi and Castellote 2006). Interoperability guarantees the capability to include many different models to simulate complex heterogeneous networks and entities with their interactions and operations. The hybrid simulation combines discrete event stochastic simulator with continuous simulators (Banks et al. 1996; Zacharewicz et al. 2008; Bruzzone 2016e). To reproduce the complexity of autonomous systems and to automate the actor behaviors, JESSI adopts intelligent agents to create a distributed control of the autonomous systems (Feddema et al. 2002; Ören and Yilmaz 2009; Bruzzone 2013e). The Intelligent Agents (IA) used in this case are based on Intelligent Agent Computer Generated Forces (IA-CGF) originally developed to reproduce social networks, human factors, and autonomous system behaviors (Bruzzone et al. 2011b; Bruzzone 2013a). One interesting aspect of this scenario is the necessity to simulate the collaborative operations involving different types of AUV, USV, and UAV with surface and underwater vessels as well as with Sensor Networks. In general, it is useful, as in JESSSI, to guarantee the use multiple RTIs (Run Time Infrastructures) including Pitch, Mäk, and Portico for a wide spectrum of integration possibilities. In the above presented case, the federation was tested with several federates and by adapting different kinds of Federation Object Model (FOM) such as SIMCJOH FOM, MCWS-MSTPA FOM, RPR (Real-time Platform-level Reference for integrating DIS legacy systems) FOM and STANAG 4684 (Virtual Ship) proposal (Bruzzone et al. 2015a). In the new generation of marine interoperable simulation, it is important to provide the opportunity for introducing new assets within the scenario as well as models characterized by different levels of fidelity based on the simulation objectives and data availability. This allows a flexible approach for experimentation and investigating over a wide spectrum of problems related to the operational use of autonomous systems and to find new ways to use them in different scenarios. It was possible to guarantee the engagement of SMEs by providing an intuitive and comprehensive representation of the EMF as an immersive virtual world. This approach allowed introduction of an innovative concept of man-in-the-loop (Magrassi 2013). This means that SMEs immersed in the virtual environment are able to capture the maritime picture and scenario awareness: thanks to 3D representation of assets and their capabilities. Sensor and weapon ranges dynamically denoted as spheres, highlight communication assets, and networks status relating directly to the evolution of the MoM. Moreover, the SME could test different hypotheses by interactively assigning high level tasks to the IAs that are controlling the traditional assets and the autonomous systems. This concept allows conducting the experimentation autonomously, avoiding the need to remotely control these assets, or to involve other people in operational details or through predefined assignments, scripts, or waypoints (Cooke et al. 2006). This approach leads toward the possibility of repeating multiple simulations and to develop new collaborative orders and behaviors through combined task assignment to UxV and traditional assets (Bruzzone et al. 2013c; Ferrandez et al. 2013; Kalra et al. 2007; Vail et al. 2003). For instance, it results in assigning a Small-Waterplane-Area Twin Hull (SWATH USV) a recovery, recharging, and fast data transfer task to cover an area for AUV deployment. In this framework, the IAs are in charge of creating a dynamic heterogeneous network that is able to self-reorganize to cover underwater communication. Indeed such capability is crucial, being very well known that in future scenarios the collaborative tasks over multiple domains will be very important to guarantee new capabilities (Fig. 14.2) (Richards et al. 2002; Ross et al. 2006; Tanner et al. 2007).

This is a good example where these simulation capabilities lead to improve the capacity of the training to address a complex scenario and be tuned and validated. This kind of simulation further improves E&T and represents an open distributed simulation framework that is able to provide a clear understanding of the events and mission environment. In fact, the users are able to figure out the scenario evolution in an intuitive way and it becomes easy to analyze the different Courses of Actions (COAs) during the Joint Naval Training through the interoperable stochastic simulation.

In this way, it is possible to evaluate the scenario dynamics as well as the consequences, in terms of MoM, of the different alternative decisions and related risks. The virtual experience accumulated via an MS2G constitutes a very valuable support for E&T. Therefore, it is possible to test hypotheses about the bandwidth of the communications among the supersonic missiles and check if they are consistent with the need to share and renegotiate targets during the attack incorporating its maneuvering capabilities and the air defenses of the task force.

14.3 Intelligence-Driven Simulation Addressing Human Behavior Modeling

The case presented in this section deals with Agent-Driven simulation. These types of simulators are becoming very popular for their capability to reproduce scenarios with many active elements and to study complex systems and where emergent behaviors evolve from the interactions among the actors. In general, it is possible to develop agents with special characteristics to create complex systems (Hewitt 1977; Wooldridge and Jennings 1995). IAs able to act within a simulation are able to drive the evolution of the events by simulating different actors. This requires developing special challenging capabilities for these agents such as autonomy, adaptability, situatedness, sociability, composability, etc. (Sarjoughian et al. 2001b; Ören 2001; Massei et al. 2014a; Bruzzone et al. 2017).

In constructive simulation for war-gaming, this represents a big potential, especially in dealing with Human Behaviors. Due to these reasons, in the presented case, to reproduce operations dealing with civilian by agent-driven simulation, the scenarios evolved along Yugoslav Wars and consequent operations (e.g., Kosovo) suggest the necessity to address these aspects for supporting training, education, and potentially operational planning in mission environments where the population outcome is one of the most critical elements (Nation 2003; Bocca et al. 2006). These scenarios could be extended to the cases of urban disorders in reference to domestic and overseas contexts (Bruzzone and Massei 2007a). In general, in recent decades, the population was in the middle of most conflicts, and coalition forces as well as the different opponents were perceived as friends or foes by civilians based on their behavior and not on their flag. It is very interesting to outline that there is a fundamental difference between civilians and military units in these scenarios. For instance, in these cases and within a constructive simulation, a battalion (a division or a wing) has to execute clear orders, normally consistent with Rules of Engagement (RoE), logistics chain, and command chain. Vice versa, a group of civilians react based on their perceptions without task assignments and, hence, represents a challenge for simulation in CAX (Cayirci and Marincic 2009a). Someone could object that in CAX there are units of civilians used traditionally as Internally Displaced Persons (IDPs) or Refugees. Therefore, these entities are normally controlled manually by the White/Grey Cell and Civilian Advisors based on predefined Scripts and they evolve usually in a predetermined way usually with no fail risk for the trainees. In addition, these elements are normally proposed in terms of small groups or simplified behaviors in order to make it possible to manage them manually within a CAX and all the responsibility for the effective training is relying on the SME. Now SMEs are fundamentally actors, therefore, it is crucial to provide them with models, tools, and simulators to support their work and turning it into a quantitative context based on a scientific approach (Bruzzone et al. 2009a). In the presented case, the adopted simulation need to be an interoperable hybrid stochastic simulation that should be federated with other simulators to cover specific aspects of a scenario dealing with Human Behavior (Bruzzone et al. 2008b). The main goal was to model complex elements such as Riots and Civil Disorders in towns and their impact on a Theater Level Wargame in order to train the Commander and his staff during a CAX to consider the dynamics of these elements with respect to the whole scenario.

In this sense, Polyfunctional Intelligent Operational Virtual Reality Agents (PIOVRA) agents were used to drive the human behavior simulation (Bruzzone and Massei 2007a). The use of agents requires application of different AI techniques to provide autonomy, awareness, and assessment capabilities. The PIOVRA agents demonstrated their capability to reduce human personnel interoperating with the simulation system, increasing the objectivity of actions and reactions of different entities present in the battlefield (i.e., friend, foe, or neutral). Another important aspect was the capability of the IA to describe basic reasons behind a particular operational behavior (allowing verification in an indirect manner, the doctrine, tactics, and ROE) by generating realistic Military Reports to Higher Commanders Bruzzone et al. (2008b).

This is critical to develop and integrate Human Behavior Models (HBM) to reproduce population as well as their impact on military and paramilitary units and on other actors during the riots (Seck et al. 2004; Bruzzone et al. 2008a). The PIOVRA agents were conceptually designed by using G-DEVS/HLA approach and all the models were HLA native. The PIOVRA Simulation passed integration test successfully with both with DMSO RTI and IEEE1516 RTIs and was experimented with JTLS as well as with other federates (e.g., IA-CGF Entity and Units) using RTI NG Pro V4.0 based on HLA Interface Specification v1.3 (Giambiasi et al. 2001; Seck et al. 2005; Bruzzone and Massei 2007a). To succeed in simulating this complex scenario, agents were embedded with AI algorithms that were based on hybrid hierarchical models and to use fuzzy logic to evaluate their perception of the reality and related reactions (Chi et al. 1991; Zadeh and Kacprzyk 1992; Piera et al.1998; Bruzzone et al. 2008b; Latorre-Biel et al. 2014). PIOVRA was based on the first generation of IA-CGF and was able to confirm their capabilities in terms of autonomous reporting, decision-making, and reproduction of human factors. In this case, the modeling of complex human behaviors required creating specific models for stress, fear, fatigue, aggressiveness (Seck et al. 2005). Stress is a crucial component in urban warfare and civil disorders for both population and units. It corresponds to an emotional intense reaction related to external stimulation, providing adaptive, physical, and psychological reactions. If the efforts from the subject are failing, due to the fact that stress level overpass reaction capabilities, the individuals are subjected to vulnerabilities in relation to psychological, somatic, and combined disorders. PIOVRA agents do not deal with pathological situation that already corresponds to have units and people almost disabled, but focus on how this condition affects decisions and behavior. In PIOVRA, stress is regulated by algorithms derived by a simplified Lazarus and Folkman model (Folkman et al. 1986; Folkman 2013).

Copying Resources represent processes devoted to manage a situation combining both emotional and rational aspects. These reactions are dependent on the parameters of the PIOVRA Agents characterizing the status of units and people defined as Action Object. The value of action object parameters, dynamically evolve during the simulation and drive the different alternative behaviors against the same stressing events. In PIOVRA, each Action Object reacts to events based on these models considering the dynamic evolution of stress level. The stress is incremented based on an algorithm that consider its temporal evolution by introducing a hysteresis phenomena and is influenced by the relative perception of stressful events for a specific Action Object. Stress level of an Action Object usually has to be compared with Action Object Stress Tolerance Capability (FTC) in order to compute its effect on the different possible actions. Figure 14.3 proposes an example of PIOVRA Stress Model. The Action Objects, corresponding to the entities that interact with the Comportment Objects on the terrain, and deal with the different groups and organizations (e.g., ethnic groups, police, etc.) and influence other Action Objects belonging to the same party.

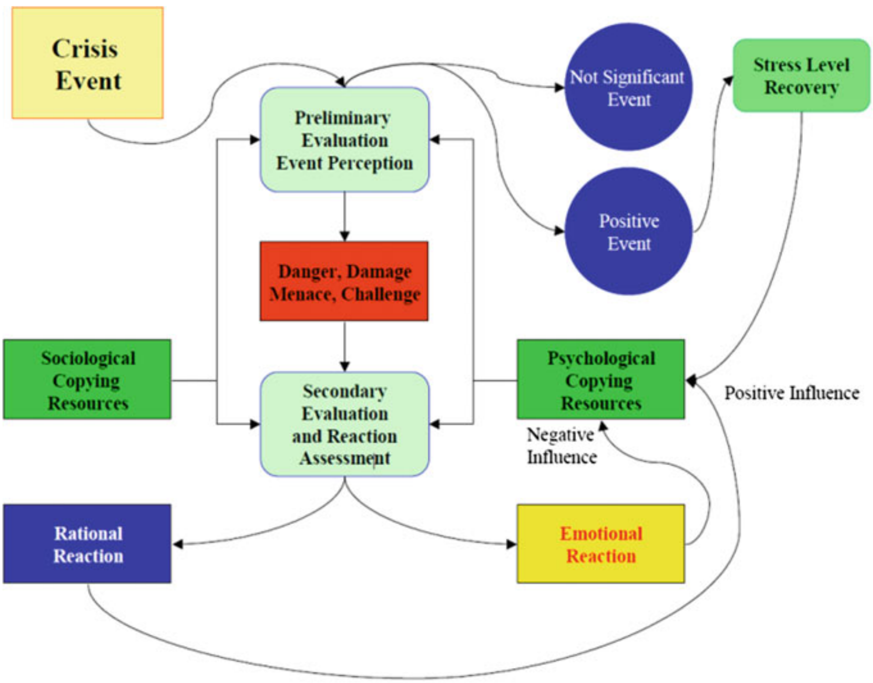


Fig. 14.3 Stress Model if IA-CGF

14.4 Crime as an Intriguing Element in Simulation During Normalization Missions

In many modern scenarios where simulation is used to support war-gaming, the traditional force-to-force game is not common. The new warfare paradigm evolved in recent year included asymmetric and hybrid warfare. Therefore, in addition to these conditions, often, there are situations where an intriguing and critical component is provided by actors that are strongly influential, quite relevant, and must be taken into consideration for mission success. The Criminal Organizations and Warlords represent a very good example of these entities.

Hereafter, we now present the case of Riots, Agitators, and Terrorists by Simulation (RATS) applied to an urban environment, during the normalization of a region, where different warlords are active and could be driven by intelligent CGF (Bruzzone et al. 2008a, 2014a). In this case, a crucial element is represented by identifying the necessary parameters to measure the capability of the agents for reproducing the behavior of the warlords and criminal organizations. For each of these agents, a "starting state" and a desired "final state" was defined to drive the capability of higher level agents, representing one of the warlords that assigns orders to its agent-driven units that achieve their goals: to react to blue force action

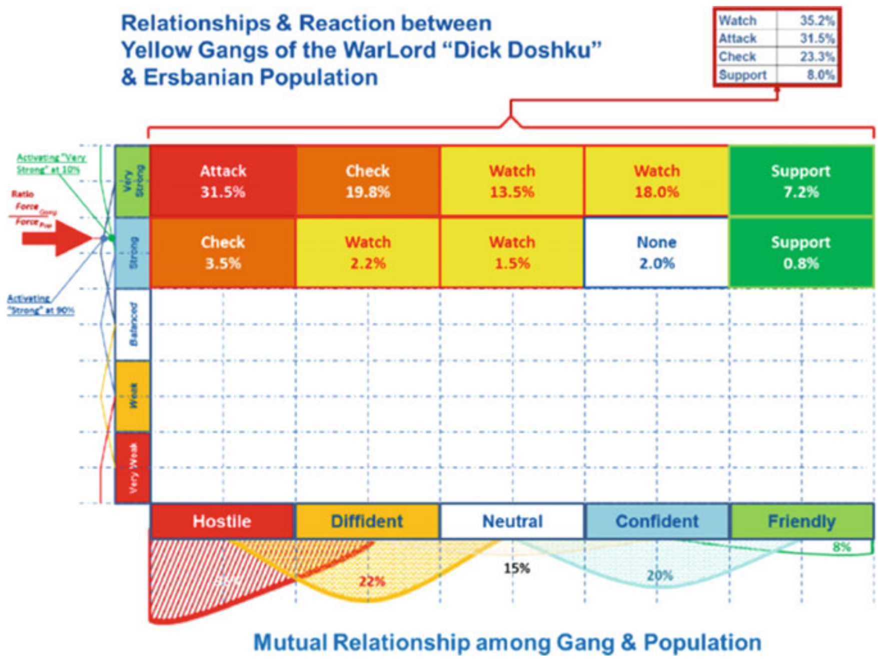


Fig. 14.4 HBM based on Fuzzy Rules

as well as to carry out routing operations (Massei and Tremori 2014b). RATS controls the movements of these (background) entities present on the area of operation of blue forces and defines the ROE of different parties with respect to each other (e.g., avoid contact, hide, engage, force escalation criteria). In general, models of ROE should address several aspects including criteria to drive the reactions (Shinseki 2001) such as Hostility Criteria, Scale of Force, Alert Conditions, Approval to Use Weapons, Eyes on Target, Terrain Restraints, Manpower Restrictions, Restrictions on Point Targets, etc. In this case, the ROEs are activated on the perception of the opponent. The attitudes between them are computed by employing Fuzzy Allocation Matrices (FAMs) and fuzzy relationships as proposed in Fig. 14.4 (Bruzzone et al. 2008a, 2014a).

14.5 Disaster Relief and Military Training

The above mentioned cases represent challenging examples that are crucial in preparing Commanders for current missions. Another very critical case is represented by Disaster Relief and Humanitarian Crisis (Kovács and Spens 2007). In reference to this context, we present the case of an important capability demonstration carried out for the first time in ITEC, London, just two months after the big

Fig. 14.5 Demonstrating the Joint Haiti Earthquake Simulation

Earthquake in Haiti Magnitude 7.0 Mw with over 100,000 deaths (Kolbe et al. 2010). The US Joint Force Command, in collaboration with several institutions and companies, was able to present the potential for federation training in HLA multilevel different systems for this purpose (Grom et al. 2010; Bruzzone et al. 2011e; Massei et al. 2014a). Specifically, the following systems were involved in this simulation (Fig. 14.5):

- DI-GUI: Representing People and Crowd by Virtual Humans in Port Au Prince
- IA-CGF NCF EQ: Simulating the Population Behavior and the effects of the Earthquake on them and Food Distribution Operations
- IA-CGF NCF: Riot Using IA to drive the looters
- JTLS: Simulating Logistics Support provided to the Area of the Disaster
- JCATS: Simulating Tactical Operation in Port Au Prince for guaranteeing security in the area
- PLEXIS: Flight Simulator allowing to fly over the Port Au Prince area
- VBS/2: Allowing to Control in First Person a Dismounted Soldier in the destroyed area.

This example confirms the importance of combining interoperable simulation and IA to support development planning. The use of simulation represents a strong benefit to improve planning of infrastructures and plants devoted to disaster relief. In these scenarios, it is very useful to federate HBM including their social networks to understand the impact on the population of the relief operations (Massei et al. 2014a). The additional seismic storms provide a challenge to the planner, requiring proper planning of its operations, and to adapt to the dynamic evolution of the crisis. These types of simulations are available today for their use in combined exercises between Armed Forces and Civil Agencies for addressing Crisis Management as forecasted many years ago (McLeod and McLeod 1984). Indeed, these considerations outline the opportunity to develop joint training opportunities for people devoted to guarantee interoperability among civil organization and military units in this sector.

14.6 DIES-IRAE: IDP & Refugees as a Challenge for Military Planners

In addition to crisis generated by disasters, another emerging issue for Military Commanders and Planner is to face problems related to IDP, Refugees and in general, with Humanitarian Crisis and their impact on the migration phenomena (Abi-Saab 1978; Levine et al. 1985; Samers 2004; Ratha et al. 2016). In this case, we present a modeling approach to represent the complex reality of these problems. The demographics and generic statistics are used as input data, while human behavior models are in charge to represent the phenomena as well as the

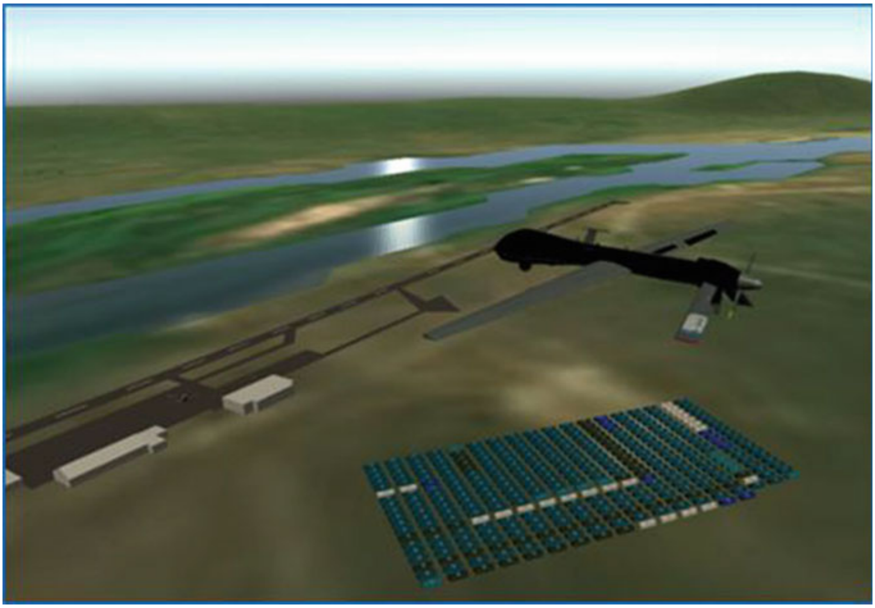


Fig. 14.6 UAV interoperating with IDP Dynamics

interactions among the different key factors and players (for example, ACLED; DESA, IOM, UNHCR). The development of conceptual models, addressing the migration flows due to wars, is crucial to properly plan future operations that affect huge amount of refugees and IDP (Fig. 14.6) (Bonabeau 2002; Anderson et al. 2007; Bruzzone and Sokolowski 2012b; Sokolowski and Banks 2014; Bozzoli et al. 2015).

Disasters, Incidents, and Emergencies Simulation Interoperable Relief Advanced Evaluator (DIES-IRAE) is an example of multilayer approach that is able to combine different M&S techniques including stochastic discrete event constructive simulation, virtual simulation, and agent-driven simulation (Bruzzone et al. 2016b). The simulation was developed in strong relationship with NATO M&S COE and validated over different scenarios in Africa. The system combines a Virtual and a Constructive simulator integrated in HLA.

14.7 CIMIC and PSYOPS: Operational Planning Versus Human Behavior Modeling

We already mentioned that while it is very common to use simulation for running wargames, in the last 20 years overseas operations of most coalitions (e.g., UN, NATO) required developing new skills. For instance, in stabilization or normalization scenarios, the Civil Military Cooperation (CIMIC) and the Psychological Operations (PSYOPS) are very fundamental activities. These activities are further stressed in new emerging scenarios (Pingitore 2004; Ankersen 2007; Fishstein and Wilder 2012). This requires developing training programs and consideration of the impact of these operations on the population. This context is very challenging as it has to address both the HBM and the information model of targets, interest groups, etc. The case presented here deals with CIMIC and Planning Research. In Complex Operational Realistic Network (CAPRICORN) simulation developed within European Defense Agency (EDA), the development of this innovative simulation system was led by Simulation Team in cooperation with DIME/DIPTEM, LSIS, MAST, and ITESOFT and with support of EDA and European National MoD (i.e., Italian and French Ministry of Defense). CAPRICORN allowed further development of IAs for reproducing population beliefs evolution by creating a multilayer structure. CAPRICORN simulator operates within an HLA Federation extended to other simulators and supports E&T for operational planners in CIMIC and PSYOPS. These elements were integrated within the whole planning system for adopting a comprehensive approach over the whole scenario (Fig. 14.7) (Bruzzone 2013a).

The model allows definition of configuration groups that generate the whole population, the social networks, and their deployment on the terrain (e.g., houses, work places, etc.). The people-object is characterized in terms of Age, Gender, Ethnics, Tribe, Social Status, Education, Religion, Political Party, Life Cycle, etc. In addition, each people-object has human behavior modifiers and connections to multiple interest groups (e.g., a Leader, a Village, a Church, a Business Sector)

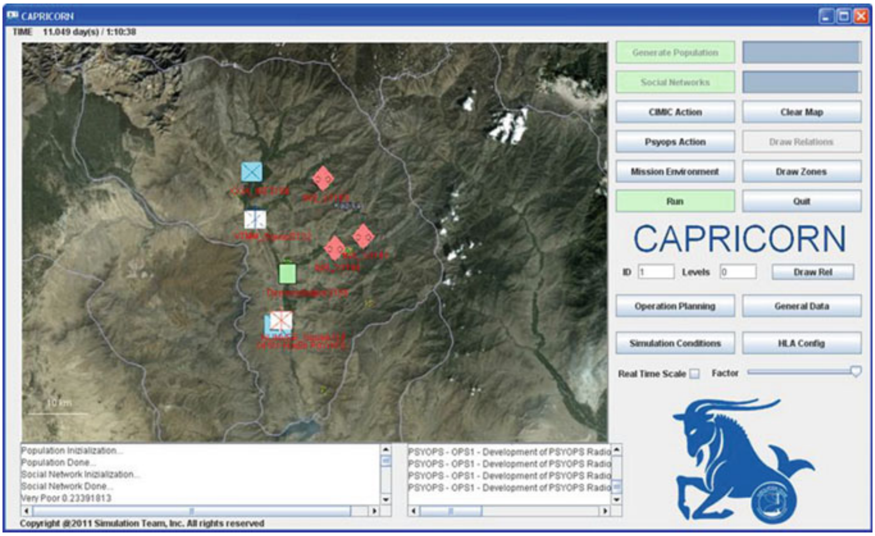


Fig. 14.7 CIMIC & PSYOPS Simulation

(Fig. 14.8). In this way, CAPRICORN allows evolution dynamically as the interrelationships evolve at individual and group levels based on the effect of the CIMIC, PSYOPS, and other operations.

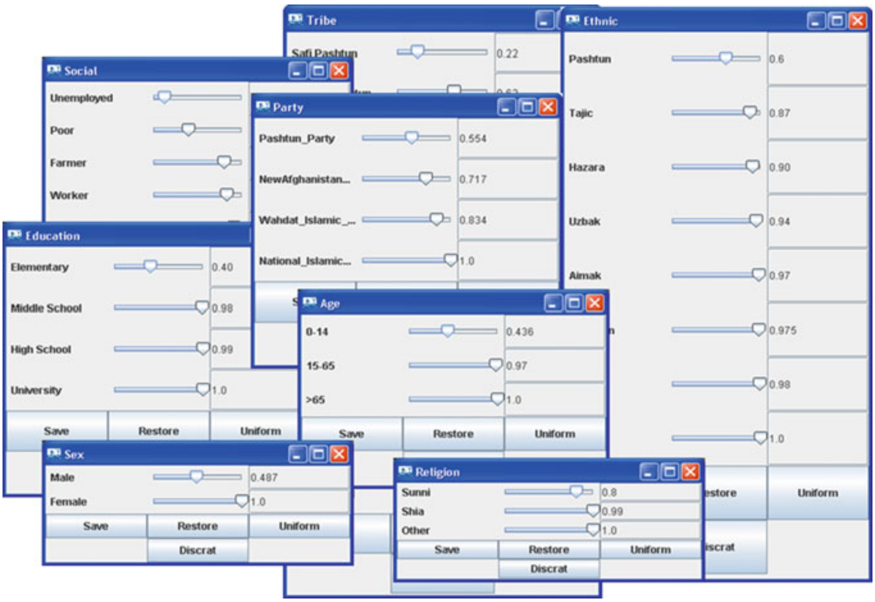


Fig. 14.8 CAPRICORN Population Group Definition

14.8 Education on Information-Sharing and Training for Intelligence

Another important area for E&T is the communications sector from HUMINT to STRATCOM. In this area, the traditional education programs could heavily benefit by utilizing M&S. In particular, SG are very promising in this sector and the proposed case of Sibilla is a good example of the potential of these solutions and their capability for different purposes (Bruzzone et al. 2009d). Sibilla simulator originated through a joint cooperation between NASA and academic institutions to transform a RPG board game to support team building, learning group dynamics, and information management into an SG (Nylen et al. 1967; Elfrey 1982). In this web game, the participants have to face a crisis dealing with multiple potential terrorist attacks. There is no time for players to learn about one another, to go slow and consolidate relationships, or consider interpersonal relations such as the standard perspectives of inclusion, control, and affection. The terrorist organizations are controlled by intelligent agents that take care of reproducing the processes of planning, preparing, and executing attacks. Along these phases, there are spills of information that are distributed among the players often in incomplete ways to provide parts of the information about the incoming events. The players could decide to go through data mining of this information or to share it with others. This is an example where data fusion has to be applied to information instead than to sensor data (Longo 2012). Based on the different E&T scenarios, SG is used for training users in remote classroom operating on PCs, or even smart phones, and provides capability to create a progressive evolution of the trainees in terms of understanding the problem and learning additional skills over more challenging scenarios. Indeed, the web technologies allow distributing the game and emphasize E&T effectiveness through remotely controlled training sessions.

14.9 Commander Strategic Decision-Making: Engagement into Serious Games

Preparing a Commander for a new task is a challenge, especially because his capabilities and skills are usually already pretty consolidated and based on long experience acquired directly on the field. So, it is hard to develop any Strategic Decision-Making course for a Commander considering his background, experience, attitude as well as the skills that made him a Commander (Elfrey 1982). This does not mean that Military Training should ignore the fundamental aspect to introduce Commanders in Strategic Decision-Making.

In fact, today the scenarios change and evolve so quickly that the experience acquired by Commanders need to be adapted and extended over new areas (e.g., different geopolitical regions or new domains). This is a great opportunity for innovative Simulation and SGs. It is important to outline that there is a special interest with respect to the impact of strategic decision involving human factors within overseas scenarios (Main 2009). Within these contexts, the human factors are often the main aspects to be addressed, as happened in recent scenarios such as Libya, Afghanistan, Syria (Johnson and Mason 2008; Kreps 2010; Bellamy and Williams 2011; Dewachi et al. 2014).

The presented case relates to a new concept derived from SG and MS2G: Simulation of Multi Coalition Joint Operations involving Human Modeling (SIMCJOH) is an MS2G, devoted to immersive experience and focusing on engaging the Commander and his staff into a time sensitive and stressing situation related to strategic issues (Bruzzone et al. 2015d) as proposed in Figs. 14.9 and 14.10.

SIMCJOH has been developed as an HLA Federation combining different systems such as SIMCJOH Virtual Interoperable Simulator (VIS) and SIMCJOH Virtual Interoperable Commander (VIC), Scenario Generator and Animator (SGA), GESI, Network Communication Simulator (NCS), etc. SIMCJOH project was conducted in cooperation between Simulation Team, DIME, Genoa University, CAE, Cal-Tek, MAST, MSC-LES University of Calabria, Leonardo, and it was cosponsored by Italian MoD. SIMCJOH has been adopted by NATO M&S Center of Excellence that supported the VV&A of the simulator and the experimentation working side by side with different Military Organizations (Di Bella 2015). Analyses were also conducted for education and training in relation to its integration with Generalized Intelligent Framework for Tutoring (GIFT) that represents an innovative intelligent tutoring system of US Army Research Lab (Sottilaire 2012b).

Fig. 14.9 SIMCJOH VIC: Virtual Negotiation

Fig. 14.10 SIMCJOH VIS Constructive Simulation

14.10 Terrorism as an Emergent Topic: From Training to Crowdsourcing

Defense against terrorism is traditionally an issue for many Agencies and Services, while Military Users have a limited role in this context. The evolution of the global situation is changing this aspect. For instance, it is over 10 years that NATO activated programs specifically devoted to Defense Against Terrorism (e.g., NATO PoW DAT). The idea to create simulation models for anti-terrorism (and even in asymmetric conflicts) has been investigated for many years and has been very popular since September 11 event (Smith 2002; Abrahams 2005; Ören and Longo 2008). The case study presented in this section is related to the NATO PoW DAT initiative developed by NATO HQs, in cooperation with STO CMRE and focused on developing an SG that is able to support training, but also to address Crowdsourcing (Bruzzone et al. 2015c). Crowdsourcing is an interesting concept that allows multiple people to contribute to finding a solution to complex problems (Brabham 2008; Massei et al. 2014a). For instance, DARPA conducted experiments on an SG to collect and analyze most promising tactics for Anti Submarine Warfare (ASW) using new generation USV by involving web users (Dillow 2011). By engaging people through SG and by involving a large number of human players, it became possible to explore wide experimental ranges and to learn about best strategies or solutions in defense and business (Bruzzone et al. 2013d; Boinodiris and Fingar 2014).

Fig. 14.11 DVx2 Virtual Representations of Terrorist Attacks, Countermeasures

This approach could be further reinforced by Modeling and Simulation as a Service (MSaaS) concept that allows transforming simulation as a cloud service easily accessible by a wide number of people (Siegfried et al. 2014). In this case, the point was not to identify winning solutions, but to learn about assessments achieved in Defense against terrorism within NATO Projects, so different users were enabled to play the scenarios within Distributed Virtual experience and experimentation (DVx2) framework (Fig. 14.11).

14.11 Asymmetric Warfare: Marine Domain as Opportunity to Teach Agile Command and Control (C2) Concept

The Agile C2 is an innovative concept for modern scenarios dealing with the necessity to adapt the Net Centric Command and Control Maturity Models (NEC C2 M2) to the dynamic evolution of the mission environment (Alberts et al. 2014). This problem is decomposed into many cases, including complex piracy scenarios already addressed by several models (Chee et al. 2007; Xiao et al. 2009; Venek 2010; Bruzzone et al. 2011c). The case presented here, uses simulation to analyze a complex maritime scenario to evaluate alternative strategies related to different NEC C2 M2 and to understand the Agility Concept (Alberts and Hayes 2003; Bruzzone et al. 2011d). Netcentric C2 has been developed over the years to evaluate different performance and critical issues (Daly and Tolk 2003).

Piracy Asymmetric Naval Operation Patterns modeling for Education and Analysis (PANOPEA) is a stochastic discrete event constructive simulator devoted to reproduce different warfare types (e.g., traditional, asymmetric, etc.) focusing on marine interdiction with respect to C2 Agility. The case used for simulation of C2 Agility was applied to Aden Gulf scenario related to piracy. This scenario was devoted to supporting different experimentation, educational, and training purposes. It is interesting to note that the results of PANOPEA scenario contributed to the development of the conceptual analysis on C2 Agility jointly with research carried out on other valuable simulation environments covering other areas. For instance, a multiplayer experimentation based on Experimental Laboratory for the Investigation of Collaboration, Information-Sharing, and Trust (ELICIT) by US DoD Command and Control Research Program (CCRP) was developed as an outcome.

Another experimentation was carried out with respect to a failing state, with years of civil war as well as a conflict with a neighbor country, by IMAGE simulation tool developed by Defense Research Development Center (DRDC) (Lizotte et al. 2013, 2014; Bernier 2012). The Experimentation was conducted on Wargame Infrastructure and Simulation Environment (WISE) involving a land battle group

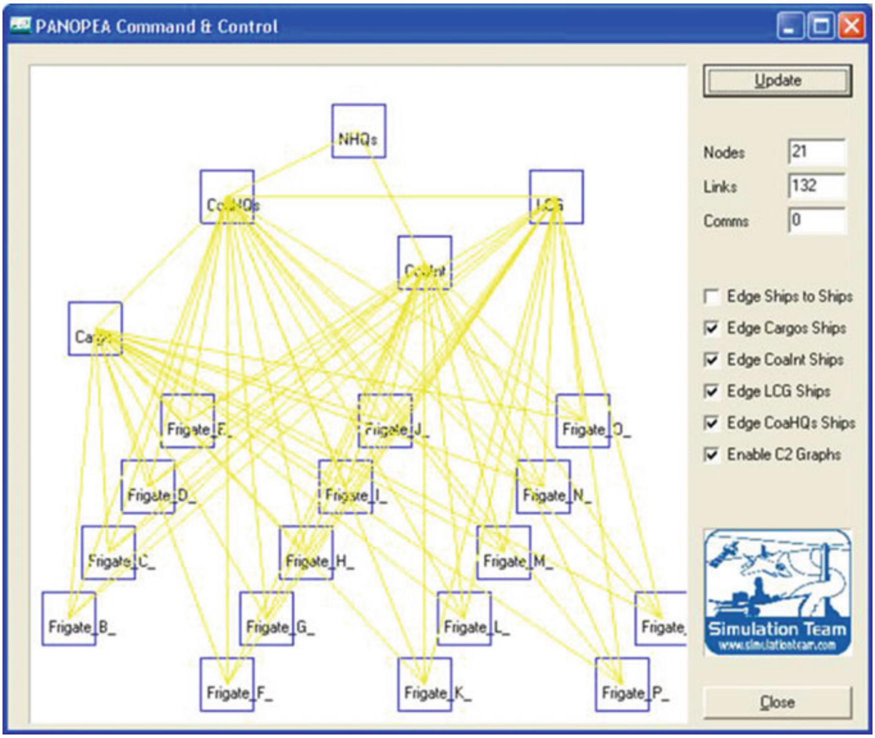


Fig. 14.12 C2 Basic Architecture

simulation at system level, including C2 models of other domains (Pearce et al. 2003). The combination of these results and the related analyses allowed extending the validity of C2 Agility concept and resulted in the best scientific achievement for C2 in 2014 for NATO Science & Technology Organization, confirming the strategic value of the stochastic agent-driven simulation (Alberts et al. 2014) (Fig. 14.12).

14.12 Moving Warfare from Asymmetric to Hybrid: From IED to STRATCOM and Cyber Attacks

Few years ago, the main issue in overseas scenarios was Improvised Explosive Devices (IED). In recent years, we have become used to see cyber and media attacks ongoing almost every day and often getting big impact in the international arena (Theohary 2011; Cameron and Putin 2013; Roth 2016). The evolution of Internet and media channels, as well as globalization emphasizes the impact of specific concurrent actions carried over different layers (e.g., political, social, financial, cyber, critical infrastructures, etc.). Due to these reasons, major players are intensely studying these new phenomena that are often aggregated under the name of Hybrid Warfare, even if this term is still controversial (Baker 2015). Strategic Communications (STRATCOM) as well as cyber attacks are among the most important new streams to conduct operations (Keeton and McCann 2005; Dimitriu 2012; Holmqvist 2013).

The basic concept is quite mature and has been used in historical cases (Lamb and Stipanovich 2016). The new technologies and media channels are generating totally new opportunities in this sector (Gerasimov 2013, 2016). The Hybrid Warfare often focuses on destabilizing Command Chains and complicating the decision-making. These actions are especially effective against organizations that are slow in their decision process due to their democratic or multinational nature. Human behavior models and the message diffusion are key elements that need to be considered. Specific models should be developed for this purpose (Faucher 2011). In this context, the subjects of war activities are often intentionally ambiguous in order to avoid a direct military confrontation. The modern concept of Hybrid Warfare is complex and requires specific models and studies. Asymmetric, Information, and Cyber Warfare evolve in critical domains considering that the Hybrid Warfare involves the whole Diplomatic/Political, Information, Military, Economic, Financial, Intelligence, legal (DIMEFIL) spectrum. In this case, the war is conducted in all battlegrounds, such as international community, home front population, and conflict zone population. Model of engagements to be adopted are the quite different. Considering the nature of this kind of warfare, it is necessary to conduct an analysis over a spectrum of alternative multiple layers (McCuen 2008; Weitz 2009; Gerasimov 2013, 2016; Bachmann and Gunneriusson 2014; Davis 2015). The actors of hybrid warfare belong to different types including both state and non-state players. Hybrid warfare is a concrete and actual phenomenon, so the development of simulation tools in this area should be addressing it to support education and training, experimentation and capability development. The use of IAs has been adapted to create a Simulation framework for this purpose (Massei and Tremori 2014b; Di Bella 2015). Threat network simulation for REactive eXperience (T-REX) was recently developed as an interoperable MS2G-based on a stochastic discrete event simulation that is able to act in stand-alone mode or federated with other HLA simulators (Fig. 14.13) (Bruzzone et al. 2016a; Bruzzone 2016d).

T-REX could be executed in real time or fast time. In the second case, it allows conducting multiple runs to investigate alternative COAs and solutions for vulnerability reduction in Hybrid Warfare. T-REX was used to support NATO MSG on Hybrid Warfare Modeling and Simulation with special attention to finalizing the M&S Requirements for this kind of warfare (Cayirci et al. 2016). The proposed simulator has already been demonstrated over a scenario related to a desertic area facing the sea where five other towns are present (see Fig. 14.13). The simulation includes multiple layers including population (e.g., individuals and/or families) as well as interest groups (e.g., industrial sectors, religious groups, social classes). These elements are structured within social networks and regulated by mutual relationships expressed by fuzzy variables in terms of attitude and intensity. T-REX includes also other layers interoperating with the socials, in particular the cyber layer and the Entity & Units (E&U) reproducing military units and assets, as the drone attacks the critical infrastructure (Fig. 14.14) influencing the scenario evolution and population behaviors.

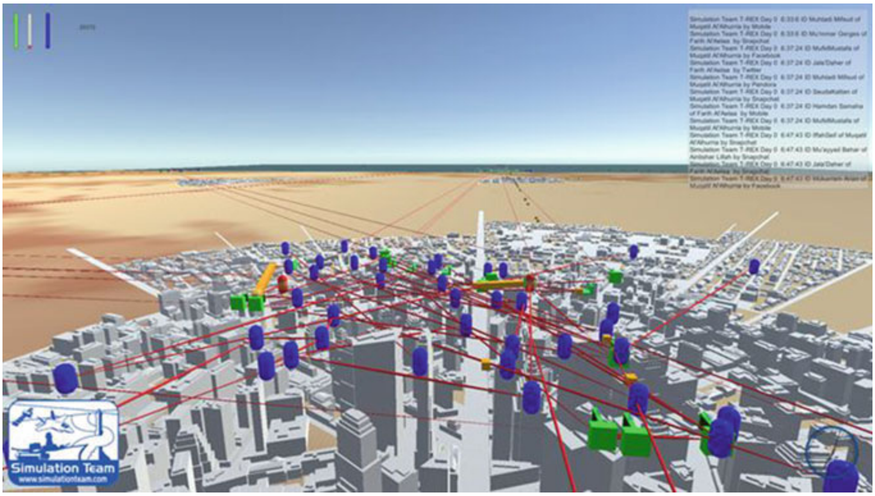


Fig. 14.13 T-REX Hybrid Warfare Simulation

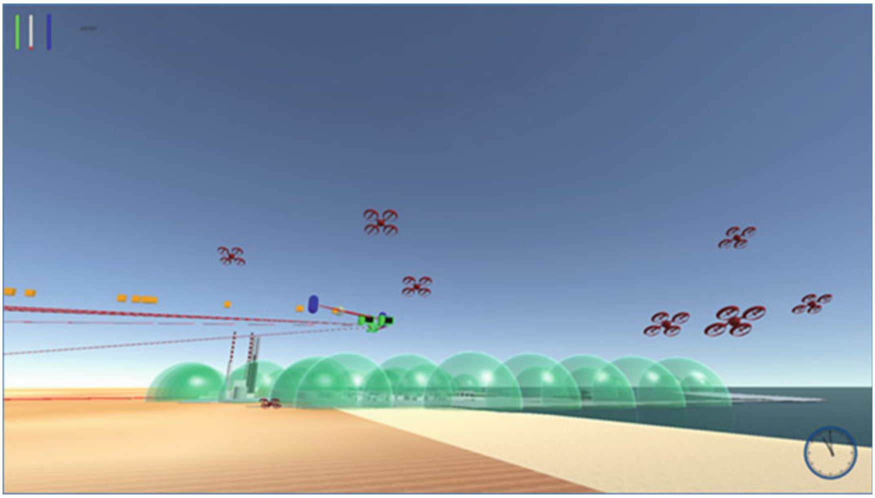


Fig. 14.14 Examples of E&U Layer integrated in T-Rex

14.13 Live Simulation: New Challenges and Opportunities

The Live simulation allows putting real equipment at work, interconnected by computers, changing the traditional exercise in something new, where casualties and damages are computed by computer algorithms that consider the dynamics of boundary and environmental conditions in detail. The major original problem that focused on creating a laser device to estimate targeting on dismounted soldiers evolved, morphed into computing more sophisticated aspects. For instance, the damages from an air drop weapon over troops or the fight between live main battle tanks and helicopters or fighters attacking a naval task force. In the last decade, special force operations and urban warfare pushed the Live simulation to address these issues by creating Military Operations on Urban Terrain (MOUT) able to represent buildings and to cover indoor live training. In this sense, new positioning devices and technologies evolved and several projects were developed by major Manufactures and MoD creating new solutions (e.g., GladiatorTM, I-MILESTM, SIAT, etc.). It is interesting to note that new Live Simulation is moving up from original single or small weapons systems up to new contexts. It becomes more and more crucial to create environments addressing joint fire control, multiple systems, communication and fire, special operations, etc.

It is important to state that several new areas are under development for live simulation. For instance, one of the most successful within the last two decades is related to Live Training in Medical Simulation, where it is possible to operate real equipment on "*simulated patients.*" In this context, the first solutions were available since beginning of '90 in major research centers such as National Center for Simulation, Institute for Simulation and Training, NHRC (Freeman et al. 1997;

Petty et al. 1999). These systems have further evolved and in the third millennium, became very sophisticated and renewed by innovative solutions such as Da Vinci (Sun et al. 2007). Another very interesting sector, currently promising new developments, is the Cyber warfare that represents a challenge for Live Simulation today. It requires enabling a specialist to be trained in these scenarios by interacting with real equipment and systems such as a Supervisory Control and Data Acquisition (SCADA) component or a Distributed Control System (DCS) component of a critical Infrastructure without affecting doing real damages (Damodaran and Couretas 2015). The case for the development initiative of an Innovative Framework for vessels that are able to face multiple aspects, even if mostly focused on Combat Management System (CMS) and tactical training: TRaining sImulation on BOaRd and distributed and for Decision-making (TRIBORDO), represents a case study moving up the capabilities of vessels by adopting innovative training solution for the crew. In this case, the focus is on the operations room with the aim to create more realistic scenarios, more easily created and executed and available to be used by trainee more times (Bruzzone 2013f). In fact, the interoperability among compatible real consoles and virtual models enables creating new simulation opportunities and new federations (Tozzi and Zini 2011).

This concept is based on the integration of Combat Operational Center (COC) and CMS on Board of vessels with different models driven by allowable agents. This approach not only allows creating task forces composed by a mix of

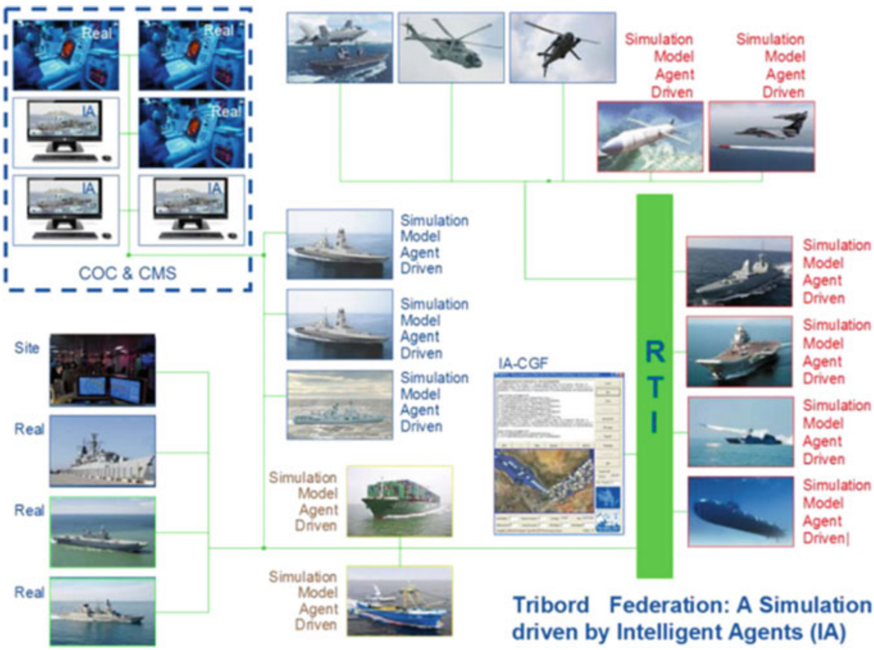


Fig. 14.15 Live Virtual and Constructive Simulation

real and virtual assets, but also allow substituting real console operators with simulated ones to populate a vessel COC by a mix of real operators and virtual ones (Bruzzone et al. 2013b) (Fig. 14.15).

It becomes possible to federate different components dealing with Live simulation such as real equipment (Console in Operations Room) with Virtual simulation such as Simulation Model Agent Driven (friendly or foe Vessel), Constructive simulation and IA-CGF (tactical simulation). In this way, a complex scenario could be created without requiring commitment and preparation of a large number of real operators. So, just a few persons will be able to prepare and conduct an exercise on their real consoles interacting with the other people of the COC and of the other ships. This will act as a force-multiplier for training opportunities, allowing conducting exercises with reduced personnel with the ship in the port. In addition, TRIBORDO allows developing new E&T programs addressing procedural and conceptual preparation of the crew and represents a clear example of how evolution of interoperable simulation is transforming the training processes using Live simulation.

14.14 Autonomous Systems and Cyber Warfare: New Areas for Training

Future of Defense is focused on two new and related issues: Autonomous Systems and Cyber Space. There is necessity to train people in dealing with these contexts and it is intuitive to understand that the related high level of complexity of these missions requires simulation (Biagini et al. 2016).

The presented case study refers to a public domain simulator experimented by NATO STO CMRE for Marine Cyber Warfare Simulation (MCWS) (Fig. 14.16). The simulator addresses heterogeneous networks with special attention to the Anti-Access/Area Denial (A2/AD) and includes multiple traditional assets, AUVs, and cyber space, playing a crucial role (Bruzzone et al. 2013e). The system has been integrated into MCWS-MSTPA Federation in HLA as well as with hardware (HIL) and software (SIL) of real autonomous systems. This Case study moves the problem from traditional game theory to simulation, to be able evaluate the impact of operations and technological details dealing with different tactics with respect to the success rate of the whole scenario (Kuhn 1997; Zeigler et al. 2000).

The model was used to investigate the potential for creating mixed scenarios including legacy systems, traditional assets (ships) and autonomous vehicles operating over the Extended Maritime Framework. In this context, communications are running over Heterogeneous Networks and are strongly dependent on Cyber Defense, so the simulator has a simple scenario, is modular and scalable. It involves a surface vessel patrolling an area by using autonomous systems in order to guarantee area denial to a potentially hostile submarine. The context includes several sensors, weapon systems, AUV, buoys, helicopters, etc. In addition, the

Fig. 14.16 MCWS Simulator

mission environment could move from A2/AD up to operational military confrontation.

It is evident that the "devil is in the details." For instance, the different bandwidths and reliabilities of the heterogeneous networks affect the effectiveness of the procedures and require investigating new approaches and policies to deal with the autonomous network. In this case, it included a new multistatic sonar solution based on AUV and buoys that could be integrated in the whole architecture (Bruzzone et al. 2013c).

14.15 Summary

This chapter presented examples as scientific evidence of the potential of M&S applied to Military Training. The presented problem cases are all in public domain, therefore are results of real projects applied to real cases. They make evident the complexity of this context as well as the capabilities of modern M&S.

The success in this area, as seen in the presented cases, is strongly related to the capability to engage experts on the operational field with simulation scientists. This aspect is crucial to develop the proper conceptual models and architectures, to implement them in a way able to support the users and to successfully complete the VV&A processes.

The proposed experiences have been deliberately focused on new emerging scenarios and innovative models and paradigms. This has been done not only to provide the reader with an up-to-date understanding of this context, but also to demonstrate that simulation evolves continuously, despite the fact that the fundamental concepts are still the same after half a century (e.g., importance of VV&A, conceptual modeling fundamentals, etc.). In some way, this could be summarized as the need to adapt the traditional principia and foundations of M&S with the innovative emerging methodologies and the new enabling technologies. By adopting this approach in developing M&S projects, it is possible to successfully address very challenging mission environments and complex problems.

Review Questions

1. Describe the advantage of introducing Human Behavior Modeling in Simulation models.
2. Could you identify specific application of Live Simulation respect Constructive and Virtual?
3. 2.1 LVC as commercial tool and the real LVC concept behind LVC simulation: describe the difference.
4. What are the advantages of Interoperability for Military Training?
5. What are the advantages for simulation for Cyber Defense?
6. What are the critical elements to model for simulating Hybrid Warfare?
7. What is the major criticality for VV&A in Military Simulation?
8. Describe the concept of Simuland.
9. HLA: Describe the motivation for adopting it.

References

Abi-Saab, G. (1978) “The United Nations Operation in the Congo, 1960–1964”, Oxford University Press, Vol. 4

Abrahams A., Boisot M., Bharathy Gnana (2005) “Simulating the Knowledge Transfer Dilemma: Lessons for Security and Counter-Terrorism”, Proceedings of SCSC, Philadelphia, NJ, July

Abt C.C., (2002) “Serious Games”, University Press of America, NYC, USA

ACLED (2016a) “All Africa Files, Version 6 (1997–2015)”, Armed Conflict Location and Event Data Project, www.acleddata.com, December

ACLED (2016b) “Country File: Nigeria (1997–2015)”, Armed Conflict Location and Event Data Project, www.acleddata.com, December

AJP-3.10.1, “NATO Psychological Operations Doctrine”, NATO/PfP unclassified

AJP-9, “NATO Civil-Military Co-Operation (CIMIC) Doctrine”, NATO/EAPC unclassified

Alberts D.S, Bernier F., Farrell P.S., Pearce P., Belanger M., Lizotte M., Joglar M., Mitchell W., Huber R.K., Bruzzone A.G., Muria A., Tremori A., Sebastiani M., Spaans M., Reitan B.K., Langsaeter T., Freire F.J.V., Manso M., Nunes P.V., Cheah M., Tan D., Barath J., Baisini C., Johansson B.J.E., Nordlander A., Dodd L., Houghton P., Henshaw M., Ball D.J., Chan K., Davis P.K., Hayes R.E., Daly-Hayes M., Phister P.W., Gregg R.J., Meuer M., Houston N. (2014) “C2 Agility”, NATO Science & Technology Organisation Technical Report, Bruxelles, Belgium

Alberts, D. S., & Manso, M. (2012) “Operationalizing and Improving C2 Agility: Lessons from Experimentation”, Proc. of the 17th ICCRTS. Fairfax, VA, USA

Alberts D.S., Hayes R.E. (2003) “Power to the Edge”, CCPR Publications, Washington D.C.

Amico V., Bruzzone A., Guha R. (2000) “Critical issues in Simulation” Proceedings of Summer Computer Simulation Conference—SCSC, Vancouver, BC, Canada, July 16—20
Anderson, J., Chaturvedi, A., & Cibulskis, M. (2007) “Simulation Tools for Developing Policies for Complex Systems: Modeling the Health and Safety of Refugee Communities”, Health Care Management Science, 10(4), pp. 331–339
Anderson, L. W., and Krathwohl, D. R. (2001) “A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives”, Longman, NYC
Ankersen, C. (2007) “Civil-Military Cooperation in Post-Conflict Operations: Emerging Theory and Practice”, Routledge, London
Bachmann, S. D. O. V., & Gunneriusson, H. (2014) “Terrorism and Cyber Attacks as Hybrid Threats: Defining a Comprehensive Approach for Countering 21st Century Threats to Global Risk and Security”, The Journal on Terrorism and Security Analysis
Baker B.D. (2015) “Hybrid Warfare with Chinese Characteristics”, The Diplomat, September
Bergeron B, (2006) “Developing Serious Games”, Charles River Media, Newton, MA
Balci O., Glasow P., Muessig P, Page E.H., Sikora J., Solick S., Youngblood S. (1996) “DoD Verification, Validation and Accreditation (VV&A) Recommended Practices Guide”, Defense Modeling and Simulation Office, Alexandria, VA, November
Balci, O. (1994) “Validation, Verification, and Testing Techniques Throughout the Life Cycle of a Simulation Study,” Ann. Oper. Res. 53, 121–173
Balci, O. (2004). Quality assessment, verification, and validation of modeling and simulation applications. Paper presented at the Proceedings—Winter Simulation Conference, 1, 122–129.
Banks J., Carson J., Nelson B. (1996) “Discrete-Event System Simulation”, Second Edition, Prentice Hall, Upper Saddle River, NJ
Bellamy, A. J., & Williams, P. D. (2011) “The New Politics of Protection? Côte d’Ivoire, Libya and the Responsibility to Protect”, International Affairs, 87(4), 825–850
Bernier, F. (2012) “Agility of C2 Approaches and Requisite Maturity in a Comprehensive Approach Context”, Proceedings of the 17th ICCRTS. Fairfax, VA, USA
Biagini, M., Forconi, S., Corona, F., Mursia, A., Ganga, L., & Battiati, F. (2016) “The Unmanned Autonomous Systems Cyberspace Arena”, Proc. of MESA, Rome, Italy, June
Binnendijk, H., & Kugler, R. (2004) “Needed-A NATO Stabilization and Reconstruction Force”, Defense Horizons, (45), 1
Bocca, E., Bruzzone, A. G., & Rocca, A. (2006) “Algorithms Devoted to Reproduce Human Modifiers in Polyfunctional Agents”, Simulation Series, 38(4), 202
Boinepalli, S., & Brown, G. (2010) “Simulation Studies of Naval Warships using the Ship Air Defence Model”, Proceedings of the Annual Simulation Technology and Training Conference, Brisbane, Australia, May, 109–114
Boinodiris, P., & Fingar, P. (2014) “Serious Games for Business: Using Gamification to Fully Engage Customers, Employees and Partners”, Meghan-Kiffer Press, Tampa, FL
Bonabeau, E. (2002) “Agent-based modeling: Methods and Techniques for Simulating Human Systems”, Proceedings of the National Academy of Sciences, 99 (supplement 3), 7280–7287
Bozzoli, C., Brück, T., & Muhumuza, T. (2015) “Activity Choices of Internally Displaced Persons And Returnees: Quantitative Survey Evidence from Post-War Northern Uganda”, Bulletin of Economic Research, August 17
Brabham, D. C. (2008) “Crowdsourcing as a Model for Problem Solving: An Introduction and Cases”, Convergence, 14(1), 75–90
Bradford, W. J. (1992) “The Theoretical Layered Air-Defence Capability of a Ship Engaged Against Multiple Anti-Ship Capable Missile Attacks”, Technical Report of Aeronautical Research Labs, Melbourne, Australia
Bruzzone A.G., Massei M., Di Matteo R. (2017) “Innovative Modelling of Social Networks for reproducing complex Scenarios”, Proc. of I3M, Barcelona, Spain
Bruzzone A.G., Massei M., Longo F., Cayirci E., di Bella P., Maglione G.L., Di Matteo R. (2016a) “Simulation Models for Hybrid Warfare and Population Simulation”, Proc. of NATO Symposium on Ready for the Predictable, Prepared for the Unexpected, M&S for Collective Defence in Hybrid Environments and Hybrid Conflicts, Bucharest, Romania, October 17–21

Bruzzone A.G., Massei M., Di Matteo R., Agresta M., Franzinetti G., Porro P. (2016b) “Modeling, Interoperable Simulation & Serious Games as an Innovative Approach for Disaster Relief”, Proc.of I3M, Larnaca, Cyprus, September 26–28

Bruzzone A.G., Massei M., Di Matteo R., Maglione G.L. (2016c) “Model of Advanced Planner for Interoperable Computer Interactive Simulation”, Proceedings of I3M, Larnaca, Cyprus, September 26–28

Bruzzone A.G. (2016d) “Disaster Relief and Humanitarian support Simulation Demo”, CAX Forum, Ottobrunn, Germany, September 20–23

Bruzzone A.G. (2016e) “Simulation and Intelligent Agents for Joint Naval Training”, Invited Speech at Naval Training and Simulation, London, UK, July 6–8

Bruzzone, A. G., Longo, F., Agresta, M., Di Matteo, R., & Maglione, G. L. (2016f) “Autonomous systems for operations in critical environments”, Proc. of the Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems and Space Simulation for Planetary Space Exploration, Pasadena, CA, April

Bruzzone A.G., Massei M., Tremori A., Di Matteo R., Maglione G.L., (2015a) “Synthetic Environment for Interoperable Advanced Marine Simulation”, Proc. of EMSS, Bergeggi, Italy, September

Bruzzone A.G., Massei, M., Piccini, I., Tremori, A., (2015b) “Modeling Human Behaviors in Marine Environment for Anti-Piracy Operations”, Proc. of MAS, Bergeggi, Italy, September

Bruzzone A.G., Massei M., Tremori A., Camponeschi M., Nicoletti L., Di Matteo R., Franzinetti G. (2015c) “Distributed Virtual Simulation Supporting Defense Against Terrorism”, Proceedings of DHSS2015, Bergeggi, Italy, September

Bruzzone A.G., Massei M., Longo F., Nicoletti L., Di Matteo R., Maglione G., Agresta M. (2015d) “Intelligent Agents & Interoperable Simulation for Strategic Decision Making on Multicoalition Joint Operations”, Proc.of DHSS2015, Bergeggi, Italy, September

Bruzzone A.G., Massei M., Poggi S., Bartolucci C., Ferrando A. (2014a) “Intelligent Agents for Human Behavior Modeling as Support to Operations”, Simulation and Modeling Methodologies, Technologies and Applications, Advances in Intelligent Systems and Computing, Springer, London, Vol. 319, pp 119–132

Bruzzone A.G., Massei M., Agresta M., Poggi S., Camponeschi F., Camponeschi M. (2014b) “Addressing Strategic Challenges on Mega Cities through MS2G”, Proceedings of MAS, Bordeaux, France, September 12–14

Bruzzone A.G., Massei M., Tremori A., Longo F., Nicoletti L., Poggi S., Bartolucci C., Picco E., Poggio G. (2014c) “MS2G: Simulation as a Service for Data Mining and Crowd Sourcing in Vulnerability Reduction”, Proceedings of WAMS, Istanbul, Turkey, September 16–17

Bruzzone A.G., Massei, M., Longo, F., Poggi, S., Agresta, M., Bartolucci, C., & Nicoletti, L. (2014d) “Human Behavior Simulation for Complex Scenarios based on Intelligent Agents”, Proceedings of the 2014 Annual Simulation Symposium, SCS, San Diego, CA, April

Bruzzone, A. G. (2013a) “Intelligent Agent-based Simulation for Supporting Operational Planning in Country Reconstruction”, International Journal of Simulation and Process Modelling, Vol.8 (2–3), 145–159

Bruzzone A.G., Fontaine J.G., Gough E. (2013b) “White Paper on Modelling and Simulation within Maritime Research and Experimentation”, NATO STO CMRE, La Spezia, December

Bruzzone A.G., Berni A., Fontaine J., Cignoni A., Massei M., Tremori A., Dallorto M., Ferrando A. (2013c) “Virtual Framework for Testing/Experiencing Potential of Collaborative Autonomous Systems”, Proc. of I/ITSEC, Orlando, FL, November

Bruzzone, A., Mursia, A., & Turi, M. (2013d) “Degraded Operational Environment: Integration of Social Network Infrastructure Concept in A Traditional Military C2 System”, Proc. of NATO CAX Forum, Rome, Italy, October

Bruzzone A.G., Merani D., Massei M., Tremori A., Bartolucci C., Ferrando A. (2013e) “Modeling Cyber Warfare in Heterogeneous Networks for Protection of Infrastructures and Operations”, Proc. of I3M 2013, Athens, Greece, September

Bruzzone A.G. (2013f) “New Challenges for Modelling & Simulation in Maritime Domain”, Keynote Speech at SpringSim 2013, San Diego, CA, April

Bruzzone A.G., Longo F., Cayirci E., Buck W. (2012a) “International Workshop on Applied Modeling and Simulation”, Genoa University Press, Rome, Italy, September

Bruzzone A., Sokolowski J. (2012b) “Internally Displaced Persons (IDPs), Refugees & Immigrants as Agents and Models for Simulation Scenarios”, Proc. of 7th NATO CAX Forum, Rome, Italy

Bruzzone A.G., Fadda P, Fancello G., Massei M., Bocca E., Tremori A., Tarone F., D’Errico G. (2011a) “Logistics Node Simulator as an Enabler for Supply Chain Development: Innovative Portainer Simulator as the Assessment Tool for Human Factors in Port Cranes”, Simulation, October, vol. 87 no. 10, p. 857–874

Bruzzone A.G. Tremori A., Massei M. (2011b) “Adding Smart to the Mix”, Modeling Simulation & Training: The International Defense Training Journal, 3, 25–27, 2011

Bruzzone A.G., Tremori A., Merkuryev Y. (2011c) “Asymmetric Marine Warfare: PANOPEA a Piracy Simulator for Investigating New C2 Solutions”, Proc. of SCM MEMTS, June 29–30, pp. 32–49

Bruzzone A.G., Massei M., Tarone F., Madeo F. (2011d) “Integrating Intelligent Agents & AHP in a Complex System Simulation”, Proceedings of the international Symposium on the AHP, Sorrento, Italy, June

Bruzzone A., Madeo F., Tarone F., (2010) “Modelling Country Reconstruction based on Civil Military Cooperation”, Proc. of I3M2010, Fez, October

Bruzzone A.G., Frydman C., Cantice G., Massei M., Poggi S., Turi M. (2009a) “Development of Advanced Models for CIMIC for Supporting Operational Planners”, Proc. of I/ITSEC2009, Orlando, November 30–December 4

Bruzzone, A. G., Cantice, G., Morabito, G., Mursia, A., Sebastiani, M., & Tremori, A. (2009b) “CGF for NATO NEC C2 Maturity Model (N2C2M2) Evaluation”, Proceedings of I/ITSEC2009, Orlando, November

Bruzzone A.G., Tremori A., Massei M. (2009c) “Serious Games for Training and Education on Defense against Terrorism”, NATO MSG Workshop, Brussels, Belgium, October 15–19

Bruzzone A.G., Elfrey P., Cunha G., Tremori A. (2009d) “Simulation for Education in Resource Management in Homeland Security” Proceedings of SCSC2009, Istanbul, Turkey, July 2009

Bruzzone AG, Scavotti A, Massei M, Tremori A (2008a) “Metamodeling for Analyzing Scenarios of Urban Crisis and Area Stabilization by Applying Intelligent Agents”, Proceedings of EMSS, Campora SG, Italy, September 17–19

Bruzzone, A.G., Mosca, R., Bocca, E., & Massei, M. (2008b) “Critical Issues in HLA Integration: PIOVRA CGF vs. Wargaming System”, Proceedings of the Summer Computer Simulation Conference. Edinburgh, Scotland, June, pp. 213–218

Bruzzone A.G., Massei, M. (2007a) “PIOVRA Final Report”, Technical Report UNIGE, Genoa, Italy, December

Bruzzone A.G., Briano E., Bocca E., Massei M. (2007b) “Evaluation of the Impact of Different Human Factor Models on Industrial and Business Processes”, Simulation Modeling Practice and Theory, Volume 15, Issue 2, February, 199–218.

Bruzzone A.G., Cunha G.G., Landau L., Merkuryev Y. (2004) “Harbour and Maritime Simulation”, LAMCE Press, Rio de Janeriro, ISBN 85-89459-04-7.

Bruzzone A.G., Page E., Uhrmacher A. (1999) “Web-based Modelling & Simulation”, SCS International, San Francisco, ISBN 1-56555-156-7.

Cameron D., Putin V. (2013) “LIVE: David Cameron and Vladimir Putin Press Conference”, BBC, June 16 16:44BST.

Cartwright D, Zander A. (1960) “Group Dynamics: Research & Theory”, Tavistock, London, UK.

Caussanel J., Frydman C., Giambiasi N., Mosca R., Rocca A., B.M. (2007) “State of Art and Future Trends on CGF”, Proceedings of EUROSIW2007, Santa Margherita, Italy, June

Cayirci E., Bruzzone A.G., Di Bella P., Geller A. (2016) “M&S Requirements for Hybrid Warfare”, Technical Report, NATO MSG, Stavanger, Norway

Cayirci, E., & Marincic, D. (2009a) “Computer Assisted Exercises and Training: A Reference Guide”, John Wiley & Sons, NYC

Cayirci, E. (2009b) “Multi-Resolution Federations in Support of Operational and Higher Level Combined/Joint Computer Assisted Exercises”, Proc. of Winter Simulation Conference, Austin, TX, December, pp. 1787–1797

Chauvancy F. (2011) “Réflexions Militaires sur les Opérations d’influence dans le Cadre de la Stabilization”, Proc. of the Séminaire sur la Compréhension des paramètres d’environnement dans les opérations de stabilisation, Université des Sciences de Marseille, January 5

Chee Wan Ng, Arnold H. Buss, John Hiles, (2007) “Discrete-event Simulation with Agents for Modeling of Dynamic Asymmetric Threats in Maritime Security Naval” Naval Postgraduate School, Monterey, California, Thesis approved for public release

Chi, S. D., Zeigler, B. P., & Cellier, F. E. (1991) “Model-based Task Planning System for a Space Laboratory Environment”, Proc. of Fibers, International Society for Optics and Photonics, Boston, MA, February, pp. 182–193

Clausewitz von C.P.G. (1832) “Vom Kriege”, Berlin, Germany

Cooke N.J., Pringle H.L., Pedersen H.K., Connor O. (2006) “Human Factors of Remotely Operated Vehicles”, JAI Press, NY.

Daly, J. J., & Tolk, A. (2003) “Modeling and Simulation Integration with Network-Centric Command and Control Architectures”, Proc. of Fall Simulation Interoperability Workshop, Orlando, FL, September, pp. 1–10

Damodaran, S. K., & Couretas, J. M. (2015) “Cyber Modeling & Simulation for Cyber-Range Events”, Proceedings of the Conference on Summer Computer Simulation by Society for Computer Simulation International, San Diego, CA, July

Davis J.R. Jr (2015) “Continued Evolution of Hybrid Threats”, The Three Sword Magazine, N. 28, pp. 19–25

DESA (2015) “Trends in International Migrant Stock: The 2013 Revision - Migrants by Age and Sex”, Technical Report of the Department of Economic and Social Affair, United Nations, NYC

Dewachi, O., Skelton, M., Nguyen, V. K., Fouad, F. M., Sitta, G. A., Maasri, Z., & Giacaman, R. (2014) “Changing Therapeutic Geographies of the Iraqi and Syrian Wars”, the Lancet, 383 (9915), 449–457

Di Bella P. (2015) “Present and Futures Scenarios and Challenges for M&S terms of Human Behavior Modeling”, Invited Speech at I3M2015, Bergeggi, Italy, September

Dillow C. (2011) “DARPA Video Game Lets You Teach Military Software How to Hunt Submarines”, Technology in Popular Science, April 7

Dimitriu, G. R. (2012) “Winning the Story War: Strategic Communication and the Conflict in Afghanistan”, Public Relations Review, 38(2), 195–207

DoD (1996) “5000.61 Verification, Validation & Accreditation”, DoD Instruction, Washington, December

DoD (1998) “DoD Modeling and Simulation (M&S) Glossary”, DoD Directive 5000.59, Washington DC, January 1998

Elfrey P. (1982) “The Hidden Agenda”, Wiley & Sons, NYC

Elfrey, P. R., Zacharewicz, G., & Ni, M. (2011). Smackdown: Adventures in simulation standards and interoperability. Paper presented at the Proceedings—Winter Simulation Conference, 3958-3962. doi:10.1109/WSC.2011.6148086

Elfrey P., Bruzzone A.G., Mosca R., Naamane A., Frydman C., Giambiasi N., Neumann G., Ziems D., Capasso S., Signorile R., Ghosh, S., Nisanci A., B.C.B.M. (2001) “Logistics & Education in IEPAL International Cooperation”, Proceedings of HMS2001, Marseille October 15–17

Faucher, C. (2011) “Modeling Psychological Messages and Their Propagation”, Proceedings of I3M, Rome, Italy.

Feddema, J.T.; Lewis, C.; Schoenwald, D.A., (2002) “Decentralized Control of Cooperative Robotic Vehicles: Theory and Application”, Robotics and Automation, IEEE Transactions on, vol. 18, no. 5, pp. 852, 864, Oct

Ferrandez J.M., De Lope H., De la Paz, F. (2013) “Social and Collaborative Robotics”, International Journal Robotics and Autonomous Systems, 61, 659–660

Filkins, D. (2010) "With US Aid, Warlord builds Afghan Empire" The New York Times, 5

Fishstein, P., & Wilder, A. (2012) "Winning Hearts and Minds? Examining the relationship between Aid and Security in Afghanistan", Feinstein International Center Technical Report, Tufts University, Medford, MA

Fishwick, P. (1995) "Simulation Model Design and Execution", Prentice Hall, New Jersey

Folkman, S. (2013) "Stress: Appraisal and Coping", Springer, NYC

Folkman, S., Lazarus, R. S., Dunkel-Schetter, C., DeLongis, A., & Gruen, R. J. (1986) "Dynamics of a Stressful Encounter: Cognitive Appraisal, Coping, and Encounter Outcomes", Journal of personality and social psychology, 50(5), 992

Freeman, K., Thompson, S., Allely, E., Sobel, A., & Stansfield, S. (1997) "A Virtual Reality Training System for the Triage and Stabilization of Head Trauma and Multiple Injury Patients", Technical Report in Naval Health Research Center, San Diego, CA

Fullford, D. (1996) "Distributed Interactive Simulation: It's Past, Present, and Future", Proc. Wintersi, Coronado, CA, pp. 179–185

Gambardella, L. M., Rizzoli, A. E., Fornara, N., Alessandrini, A., & Epifani, C. (2000) "The Use of Simulation in the Socio-Economical Evaluation of the Management of an Intermodal Terminal", Proceedings of the International Workshop HMS, Portofino-Italy, pp. 71–76

Garrido De Ceita A.L., Torres F.C., Frydman C. (2003) "Specification and Simulation of Reactive Knowledge Based Systems", Proceedings of SCSC2004, Montreal, July

Gerasimov V. (2016) "The Value of Science is in the Foresight: New Challenges Demand Rethinking the Forms and Methods of Carrying out Combat Operations", Military Review, January–February

Gerasimov V. (2013) "The Value of Science is in the Foresight", Voyenno-Promysh-lennyy Kurier online,, February 26, (original version)

Giambiasi, N., Escude, B., & Ghosh, S. (2001) "Generalized Discrete Event Simulation Of Dynamic Systems", Transactions of the Society for Computer Simulation International, 18(4), 216–229

Gibson, R., Medeiros, D. J., Sudar, A., Waite, B., & Rohrer, M. W. (2003) "Determining the ROI for Simulation: Increasing Return on Investment from Simulation", Proc. of the 35th conference on Wintersim, New Orleans, LA, December, pp. 2027–2032

Giustozzi, A., & Ullah, N. (2006) "Tribes and Warlords in Southern Afghanistan, 1980–2005", Technical Report in Crisis States Research Centre, London, UK, September

Goleman, D. (1995) "Emotional Intelligence", Bantam Books, NYC

Gozel, R. (2000) "Firepower Score Attrition Algorithms in Highly Aggregated Combat Models", Rand Technical Report, Washington D.C.

Graetz, J. M. (1981) "The Origin of Spacewar", Creative Computing, 7(8), 56–67

Grom, A., Hoover, A., Van Deman, Q., & Knox, D. (2010) "Advanced Human Modelling Behaviours: The Key to Optimising Individual Readiness and Organisational Effectiveness", RTO-MP-HFM-202 Technical Report, FKIE, Wachtberg, Germany

Gustavsson, P. M., Björkman, U., & Wemmergård, J. (2009) "LVC Aspects and Integration of Live Simulation" Proc. of Fall Simulation Interoperability Workshop, Orlando, FL, September

Hamilton John A. Jr., Nash David A., Pooch Udo W. (1996) "Distributed Simulation", CRC Press

Haugh, B. and Lichtblau, D., (2001) "An Information Technology Support Strategy for PSYOP Impact Analysis," IDA Paper P-3587, Institute for Defense Analyses, February

Hewitt, C. (1977) "Viewing Control Structures as Patterns of Passing Messages", Artificial intelligence, 8(3), 323–364

Holmqvist, C. (2013) "War 'Strategic Communication' and the Violence of Non-Recognition", Cambridge Review of International Affairs, 26(4), 631–650

Holzer T.L., Savage J.C. (2013) "Global Earthquake Fatalities and Population", Earthquake Spectra, Volume 29, No. 1, February, pages 155–175

Huan, Z., Liu, S. K., & Li, G. (2003) "Analysis of the Upgrading of Simulation System from DIS to HLA", Journal of Computer Simulation, 1, 023

IOM (2016d) “Compilation of Available Data and Information, Mixed Migration Flows in the Mediterranean and Beyond - Reporting Period Quarter 3, 2016”, Technical Report of the International Organization for Migration, Geneva, Switzerland, December
Iuppa N, Borst T, (2006) “Story and Simulations for Serious Games: Tales from the Trenches”, Focal Press, Milton, UK
Jackson, R., Vamivakas, A., Jackson, R., & Vamivakas, A. (1997) “An Overview of Hardware-In-The-Loop Simulations for Missiles”, Proc. of Modeling and Simulation Technologies Conference, New Orleans, LA, August
Johnson, T. H., & Mason, M. C. (2008) “Understanding the Taliban and insurgency in Afghanistan”, Orbis, 51(1), 71–89
Joshi R., Castellote G.P., (2006) “A Comparison and Mapping of Data Distribution Service and High-Level Architecture”, Real Time Innovations, 1–5
Kalra N., D. Ferguson and A. Stentz (2007) “A Generalized Framework for Solving Tightly-Coupled Multirobot Planning Problems”, Proc. of the IEEE International Conference on Robotics and Automation, April, pp. 3359–3364
Keeton, P., & McCann, M. (2005) “Information Operations, STRATCOM, and Public Affairs”, Military Review, 85(6), 83
Kim, J., & Zeigler, B. P. (1996) “Hierarchical Distributed Genetic Algorithms: A Fuzzy Logic Controller Design Application”, IEEE Expert, 11(3), 76–84
Kolbe, A. R., Hutson, R. A., Shannon, H., Trzcinski, E., Miles, B., Levitz, N., Marie Puccio, Leah J., Jean Roger & Muggah, R. (2010) “Mortality, Crime and Access to Basic Needs before and after the Haiti Earthquake: A Random Survey of Port-Au-Prince Households”, Medicine, conflict and survival, 26(4), 281–297
Kovács, G., & Spens, K. M. (2007) “Humanitarian Logistics in Disaster Relief Operations”, International Journal of Physical Distribution & Logistics Management, 37(2), 99–114
Krathwohl, D.R., Bloom, B.S., & Masia, B.B. (1964) “Taxonomy of Educational Objectives: Handbook II: Affective Domain”, David McKay Co., NYC
Kreps, S. (2010) “Elite Consensus as a Determinant of Alliance Cohesion: Why Public Opinion Hardly Matters for NATO-led Operations in Afghanistan”, Foreign policy analysis, 6(3), 191–215
Kuhl, F., Weatherly, R., Dahmann, J., (1999) “Creating Computer Simulation Systems: An Introduction to the High Level Architecture”, Prentice Hall, Upper Saddle River, USA
Kuhn H.W. (1997) “Classics in Game Theory”, Princeton University Press, NJ
Lamb C., Stipanovich S. (2016) “Back to Basics on Hybrid Warfare in Europe A Lesson from the Balkans”, Joint Force Quarterly, 81, March 29
Lanchester F.W (1995) “Aircraft in Warfare the Dawn of the Fourth Arm”, Lanchester Press, Forestville, CA
Latorre-Biel, J. I., Jiménez-Macías, E., & Pérez-Parte, M. (2014) “Sequence of decisions on discrete event systems modeled by Petri nets with structural alternative configurations”, Journal of Computational Science, 5(3), 387–394
Lau, D.C. (1979) “Confucius: The Analects”, Penguin, Harmondsworth, UK
Levine, D. B., Hill, K., & Warren, R. (1985) “Immigration Statistics: A story of Neglect”, National Academies
Liu, L., Li, Y. B., Wu, D., Zhang, L., & Wang, N. (2016) “Design and Developmental Research on the VV&A of Complex Simulation System”, Proc. of MATEC Web of Conferences, Vol. 63
Lizotte, M., Bernier, F., Mokhtari, M., & Boivin, E. (2013) “IMAGE Final Report: An Interactive Computer-aided Cognition Capability for C4ISR Complexity Discovery” Technical Report of Defence R&D Québec, Canada
Lofdahl, C. (2009) “Implementing Irregular Warfare Policy Using Modeling and simulation”, Proceedings of the 27th International System Dynamics Conference, Albuquerque
Loftin, R. B. (1994) “Virtual Environments for Aerospace Training”, Proc. of WESCON, Anaheim, CA, September, pp. 384–387
Longo F (2012) “Supply Chain Security: An Integrated Framework for Container Terminal Facilities”, International Journal of Simulation & Process Modelling, vol. 7, pp. 159–167

Longo, F., Bruzzone, A.G. (2005) “Modelling and Simulation applied to Security Systems”. Proceedings of Summer Computer Simulation Conference, pp. 183–188
Lorenz M. (1993) “Object-oriented Software Development: A Practical Guide”, Prentice Hall, New York
Lundquist E.H. (2009) “Aircraft Carriers Large and Small”, Defense Media Network, July 23
Magrassi C. (2013) “Education and Training: Delivering Cost Effective Readiness for Tomorrow’s Operations”, Keynote Speech at ITEC2013, Rome, May 22–24
Main, F. S. (2009) “Psychological Operations Support to Strategic Communications in Afghanistan”, Technical Report Army War College, Carlisle Barracks, PA
Mak, E., Mittal, S., Hwang, M. H., & Nutaro, J. J. (2010) “Automated Link-16 Testing Using the Discrete Event System Specification and Extensible Markup Language”, the Journal of Defense Modeling and Simulation, 7(1), 39–62
Malley, J. H. (1984) “Improving the Definition of the Army Objective Force (IDOFOR) - A Methodology Development Study”, Systems Analysis and Modeling in Defense, Springer NYC
Man Kwon Nam (1980) “An Examination of the Conceptual Basis of a Tactical, Logistical, and Air Simulation”, Thesis, NPS, Monterey, March
Martínez-Salio, J. R., Lopez-Rodriguez, J. M., Gregory, D., & Corsaro, A. (2012) “A Comparison of Simulation and Operational Architectures”, Proc. of Fall SIW, Orlando, Florida, September
Massei, M., Poggi, S., Agresta, M., & Ferrando, A. (2014a) “Development Planning Based on Interoperable Agent Driven Simulation”, Journal of Computational Science, 5(3), 395–407
Massei M., & Tremori A. (2014b) “Simulation of an Urban Environment by Using Intelligent Agents within Asymmetric Scenarios for Assessing Alternative Command and Control Network-Centric Maturity Models”, the Journal of Defense Modeling and Simulation, 11(2), 137–153
McCuen, J. J. (2008) “Hybrid Wars”, Military Review, 88 (2), 107
McGlynn L. (1996) “In Pursuit of M&S Standards”, Proceedings of ESS1996, Genoa, Italy
McLeod, J., & McLeod, S. (1984) “Simulation of the Society”, Simulation, 43(6), 320–323
McLeod, J. (1968) “Simulation: the Dynamic Modeling of Ideas And Systems with Computers”, McGraw-Hill, NYC
Michael D, Chen S, (2005) “Serious Games: Games That Educate, Train and Inform”, Course Technology PTR, Boston, MA
Milano V. (2014) “Naval M&S Activities within NATO”, Proceedings of Sim&Sea, Genoa, Italy, June
Miller, D. C., & Thorpe, J. A. (1995) “SIMNET: The Advent of Simulator Networking”, Proceedings of the IEEE, 83(8), 1114–1123
Mittal, S., Zeigler, B. P., Martín, J. L. R., Sahin, F., & Jamshidi, M. (2008) “Modeling and Simulation for Systems of Systems Engineering”, in Systems of Systems Innovation for 21st Century, Wiley & Sons, NYC
Möller, B., & Dahlin, C. (2006) “A First Look at the HLA Evolved Web Service API”, Proc. of 2006 Euro Simulation Interoperability Workshop, Stockholm, Sweden, June 19–22
Montgomery Douglas C. (2000) “Design and Analysis of Experiments”, John Wiley & Sons
Nation, R. C. (2003) “War in the Balkans, 1991–2002”, Strategic Study Institute, US Army War College Press, Carlisle Barracks, PA
NATO ACT (2009) “ACT M&S Vision”, Technical Report ACT, Document 5000 TC-701/TT-2368/Ser: NU0007, January 6
NATO (2012) “NATO Modelling and Simulation Master Plan: Version 2.0”, Technical Report NMSG, Document AC/323/NMSG(2012)-015
NATO (1998) “NATO Modelling and Simulation Master Plan: Version 1.0”, Technical Report, Document AC/323/(SGMS)D/2
Nelson, C. R. (2006) “How Should NATO Handle Stabilisation Operations and Reconstruction Efforts? Atlantic Council of the United States Policy Paper, Washington DC, September

Nylen D., Mitchell J.R., Stout A.T. (1967) "Handbook of Staff Development and Human Relations Training: Materials developed for use in Africa", NTL Institute for Applied Behavioral Science, Washington D.C.

Ören, T.I. and L. Yilmaz (2009) "On the Synergy of Simulation and Agents: An Innovation Paradigm Perspective", Special Issue on Agent-Directed Simulation. The International Journal of Intelligent Control and Systems (IJICS), Vol. 14, Nb. 1, March, 4–19

Ören T.I., Longo F., (2008) "Emergence, Anticipation and Multisimulation: Bases for Conflict Simulation", Proceedings of the 20th European Modeling and Simulation Symposium, EMSS 2008, pp. 546–555

Ören, T. I. (2001) "Impact of Data on Simulation: from Early Practices to Federated and Agent-Directed Simulations", Proc. of EUROSIM, June, pp. 26–29

Orlansky, J., Dahlman, C. J., Hammon, C. P., Metzko, J., & Taylor, H. L. (1994) "The Value of Simulation for Training", Technical Report of Institute for Defense Analyses, Alexandria, VA

Page, E. H., & Smith, R. (1998) "Introduction to Military Training Simulation: A Guide for Discrete Event Simulationists", Proc. of Winter Simulation Conference Proceedings, December, Washington D.C., Vol. 1, pp. 53–60

Panteli, N. & Duncan, E. (2004) "Trust and Temporary Virtual Teams: Alternative Explanations and Dramaturgical Relationships", Information Technology and People, 17 (4): 423–441

Pearce, P., Robinson, A., & Wright, S. (2003) "The Wargame Infrastructure and Simulation Environment (WISE)", Proc. of the Knowledge-Based Intelligent Information and Engineering Systems Conference, Oxford, UK, pp. 714–722

Petty, M. D., & Windyga, P. S. (1999) "A High Level Architecture-Based Medical Simulation System", Simulation, 73(5), 281–287

Philipps S.G. (2009) "Affective Learning in General Education", Assessment in University General Education Program, 6(1), 1–44

Piera, M. A., Gambin, A. J., & Vilanova, R. (1998) "Object Oriented Stochastic Petri Net Simulator: A Generic Kernel for Heuristic Planning Tools", Proc. of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Berlin, June, pp. 244–253

Pimental, K., and Blau, B. (1994) "Teaching Your System To Share", IEEE computer graphics and applications, 14(1), 60

Pingitore A. (2004) "Military Operations Research and Operation Iraqi Freedom", Thesis Report, Wells College

Ratha, D., Eigen-Zucchi, C., & Plaza, S. (2016) "Migration and Remittances Factbook 2016" World Bank Publications, Washington DC

Ratzenberger A.C. (1996) "The High Level Architecture and The Next Generation Combat Simulations", Proceedings of ESS1996, Genoa, Italy

Raybourn, E.M. (2012) "Beyond Serious Games: Transmedia for more Effective Training & Education", Proc. of DHSS2012, Rome, Italy

Raybourn, E., & Regan, D. (2011) "Exploring E-Portfolios and Independent Open Learner Models: Toward army learning concept", Proceedings of I/ITSEC, Orlando FL, Nov–Dec

Raybourn, E.M., Fabian, N., Tucker, E., Willis, M. (2010) "Beyond Game Effectiveness Part II: A Qualitative Study of Multi-role Experiential Learning", Proc. of I/ITSEC, Orlando, FL, November 29–December 3

Raybourn, E.M. (2009) "Beyond Game Effectiveness Part I: An Empirical Study of Multi-role Experiential Learning", Proc. of I/ITSEC, Orlando, FL, November 30–December 4

Rehse P., (2004) "CIMIC: Concepts, Definitions and Practice", Heft 136, Hamburg, June

Reno, W. (1999) "Warlord Politics and African States", Lynne Rienner Publishers, Boulder, CO

Reverberi, A. (2006) "Human Behavior Representation & Genetic Algorithms Characteristics Fitting", Invited Speech at PIOVRA Workshop, Savona, Italy, February

Richards A., J. Bellingham, M. Tillerson, and J. P. (2002) "How: Co-ordination and Control of Multiple UAVs", Proc. of the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, Aug., AIAA-2002-4588

Rietjens S.J.H., M. Bollen, (2008) “Managing Civil-Military Cooperation”, Military Strategy and Operational Art

Roland, R. J., Macedonia, M., Andrew, J., Metzger, J., & Miller, J. O. (1998) “Panel: The Future of Military Simulation”, Proc. of Wintersim, Washington DC, Vol. 1, pp. 813–817

Ross, S., D. Jacques, M. Pachter, and J. Raquet, (2006) “A Close Formation Flight Test for Automated Air Refueling,” Proceedings of ION GNSS-2006, Fort Worth, TX, Sep

Roth A. (2016) “Dutch probe: Missile brought from Russia downed Malaysia Airlines Plane over Ukraine”, Washington Post, September 28

Samers, M. (2004) “An Emerging Geopolitics of’ Illegal Immigration in the European Union”, European Journal of Migration and Law, 6(1), 27–45

Santos, E. E., Santos, E., Wilkinson, J. T., Korah, J., Kim, K., Li, D., & Yu, F. (2011) “Modeling Complex Social Scenarios Using Culturally Infused Social Networks”, Proc. of IEEE Int. Conference on Systems, Man, and Cybernetics (SMC), Anchorage, AK, October

Sarjoughian, H. S., & Zeigler, B. P. (2001a) “Towards Making Modeling & Simulation into a Discipline”, Simulation Series, 33(2), 130–135

Sarjoughian, H. S., Zeigler, B. P., & Hall, S. B. (2001b) “A Layered Modeling and Simulation Architecture for Agent-Based System Development”, Proceedings of the IEEE, 89(2), 201–213

Seck, M., Frydman C., Giambiasi N., Ören T.I., Yilmaz L. (2005) “Use of a Dynamic Personality Filter in Discrete Event Simulation of Human Behavior under Stress and Fatigue”, Foundations of Augmented Cognition, Vol 11, pp 908–917

Seck, M., Frydman, C., & Giambiasi, N. (2004) “Using DEVS for modeling and simulation of Human Behaviour”, in Artificial Intelligence and Simulation, Volume 3397 of the series Lecture Notes in Computer Science, Springer, Berlin, pp 692–698

Shafer, A.J.; Benjamin, M.R.; Leonard, J.J.; Curcio, J., (2008) “Autonomous Cooperation of Heterogeneous Platforms for Sea-based Search Tasks,” Oceans, vol., no., pp. 1, 10, 15–18 September

Shinseki E.K. (2001) “Operations”, Field Manual FM3-0, Department of the Army, Washington, DC, June

Siegfried, R., van den Berg, T. W., Cramp, A. J., & Huiskamp, W. (2014) “M&S as a Service: Expectations and Challenges”, Proc. 2014 Fall Simulation Interoperability Workshops (SIW), Orlando FL, September 8–12

Simpson, E. (1972) “The Classification of Educational Objectives in the Psychomotor Domain: The Psychomotor Domain”, Gryphon House, Washington, DC, Vol. 3

Smith, R. (2002) “Counter Terrorism Simulation: A New Breed of Federation”, Proceedings of the Spring 2002 Simulation Interoperability Workshop, Orlando, FL, March

Soller, A. (2001) “Supporting Social Interaction in an Intelligent Collaborative Learning System”, International Journal of Artificial Intelligence in Education, 12 (1), 40–62

Sokolowski, J. A., & Banks, C. M. (2014) “A Methodology for Environment and Agent Development to Model Population Displacement”, Proceedings of the Symposium on Agent Directed Simulation, Society for Computer Simulation International, San Diego, CA

Sokolowski, J. A., & Banks, C. M. (2011) “Principles of Modeling and Simulation: a Multidisciplinary Approach”, John Wiley & Sons, NYC

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012a) “The Generalized Intelligent Framework for Tutoring (GIFT)”, Technical Report Army Research Lab, Orlando, FL.

Sottilare, R. (2012b) “Considerations in the Development of an Ontology for a Generalized Intelligent Framework for Tutoring”, Proc. of the International Defense & Homeland Security Simulation Workshop, Vienna, Austria, September 2012

Sottilare, R., Holden, H., Brawner, K., & Goldberg, B. (2011) “Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for Team Training”, Proc. of I/ITSEC, Orlando FL, December

Sottilare, R. (2009) “Making a Case for Machine Perception of Trainee Affect to Aid Learning and Performance n Embedded Virtual Simulations”, NATO Research Workshop HFM-RWS-169 on Human Dimensions in Embedded Virtual Simulations, Orlando, FL

Stevens, J., & Eifert, L. (2014) "Augmented Reality Technology in US Army Training", Proceedings of the 2014 Summer Simulation Multiconference, Monterey, CA, July
Strassburger S., Schulze T., Fujimoto R. (1998) "Future Trends in Distributed Simulation and Distributed Virtual Environments: Results of a Peer Study", Proc. of the Winter Simulation Conference, Washington DC, December
Sun, L. W., Van Meer, F., Schmid, J., Bailly, Y., Thakre, A. A., & Yeung, C. K. (2007) "Advanced Da Vinci Surgical System Simulator for Surgeon Training and Operation Planning", the International Journal of Medical Robotics and Computer Assisted Surgery, 3(3), 245–251
Szczerbicka, H., Banks, J., Rogers, R. V., Ören, T. I., Sarjoughian, H. S., & Zeigler, B. P. (2000) "Conceptions of Curriculum for Simulation Education Panel", Proceedings of the 32nd Conference on Winter Simulation, Phoenix, AZ, December, pp. 1635–1644
Tanner H. G. (2007) "Switched UAV-UGV Cooperation Scheme for Target Detection", IEEE International Conference on Robotics and Automation, Roma, Italy, April, pp. 3457–3462
Theohary, C. A. (2011) "Terrorist use of the Internet: Information Operations in Cyberspace", DIANE Publishing, Darby, PA
Thompson, Mark (2013) "Costly Flight Hours", Time, 1, April 2
Thorpe, J. A., Bloedorn, G. W., Taylor, R., Miller, D. C., & Cyrus, M. (1987) "The SIMNET Network and Protocol", BBN Labs Inc. Technical Report, Cambridge MA
Thorin O.R., Ghustin B., Elfin D.D., Holler F.P., Brian O.G. (1982) "EW Simulation for Discrete Event Simulation", Proc. of SCGC, San Francisco, CA
Tolk, A., Cayirci, E., Shumaker, R., Adam, N. R., Pickl, S., Sullivan, J. A., & Waite, W. F. (2012a) "Defense and Security Applications of Modeling and Simulation-Grand Challenges and Current Efforts", Proc. of Winter Simulation Conference, Berlin, Germany, December
Tolk, A. (2012b) "Engineering Principles of Combat Modeling and Distributed Simulation", John Wiley & Sons, NYC
Tolk, A., & Muguira, J. A. (2003) "The Levels of Conceptual Interoperability Model", Proc. of the Fall Simulation Interoperability Workshop, Orlando, FL, September, pp. 1–11
Thorin O.R., Ghustin B., Elfin D.D., Holler F.P., Brian O.G. (1983) "EW Models for Discrete Event Simulation", Proc. of SCGC, San Francisco, CA
Townsend, J. R. (1999) "Defense of Naval Task Forces from Anti-Ship Missile Attack", Doctoral dissertation, NPS, Monterey, CA
Tozzi, D., & Zini, A. (2011). Naval operations' assessment through HLA based simulations. Paper presented at the RINA, Royal Institution of Naval Architects—International Conference on Computer Applications in Shipbuilding 2011, Papers, 3, 1–7
Tzu, S. (1963) "The Art of War", Clarendon Press, Oxford, UK
UNHCR (2015b) "World at War: UNHCR Global Trends, Forced Displacement in 2014", Technical Report of UNHCR, Geneva, CH, June
Vail D. and M. Veloso, (2003) "Dynamic Multi-Robot Coordination", Multi-Robot Systems: From Swarms to Intelligent Automata, Vol II, pp. 87–100
Vanek O., (2010) "Agent - Based Simulation of the Maritime Domain", 14th International Student Conference on Electrical Engineering. CVUT, Fakulta Elektrotechnicka, May
Weitz, R. (2009) "China, Russia, and the Challenge to the Global Commons", Pacific Focus, 24 (3), 271–297
Wells, H. G. (1913) "Little Wars: A Game for Boys of Twelve Years to One Hundred and Fifty and for that More Intelligent Sort of Girl Who Likes Games and Books", Illus, JR Sinclair, Frank Palmer, London, UK
Wells, H. G. (1911) "Floor Games", Frank Palmer, London, UK
Williams, Edward J (1996) "Making Simulation a Corporate Norm", Proceedings of the 1996 Summer Computer Simulation Conference, SCS, San Diego, CA, pp. 627–632
Wooldridge, M.; Jennings, N. R. (1995) "Intelligent Agents: Theory and Practice", Knowledge Engineering Review, 10 (2), 115–152

Yang, J. H., Kapolka, M., & Chung, T. H. (2013) "Autonomy balancing in a Manned-Unmanned Teaming (MUT) Swarm Attack", Robot Intelligence Technology and Applications, Springer, Berlin, pp. 561–569

Youngblood S.M., Pace D.K., Eirich P.L., Gregg D.M., E. Coolahan (2000) "Simulation Verification, Validation, and Accreditation", Johns Hopkins APL Technical Digest, Vol. 21, n. 3

Youngblood, S., Pace, D. K., & Mahadevan, S. (2010). Riskbased VV&A methodology the concept of M&S use risk. Paper presented at the Spring Simulation Interoperability Workshop 2010, 2010 Spring SIW, 453–459

Xiao Y., Zhang H., Li S., (2009) "Dynamic Data Driven Multi-Agent Simulation in Maritime Traffic," Prof of ICCAE, International Conference on Computer and Automation Engineering, Singapore, February 26–28, 234–237

Zaalberg T.W.B. (2006) "Countering Insurgent-terrorism: Why NATO Chose The Wrong Historical Foundation For CIMIC", Small Wars & Insurgencies, Vol 17, n. 4

Zacharewicz G., Frydman C., Giambiasi N. (2008) "G-DEVS/HLA Environment for Distributed Simulations of Workflows", Simulation, Vol. 84, N. 5, pp. 197–213

Zeigler, B. P., Ball, G., Cho, H., Lee, J., and Sarjoughian, H. (1999). Implementation of the DEVS Formalism over the HLA/RTI: Problems and Solutions. In 1999 Fall Simulation Interoperability Workshop

Zeigler, B. P., Praehofer, H., and Kim, T. G. (2000). Theory of Modeling and Simulation: Second Edition. Academic Press, San Diego, CA

L.A. Zadeh, J. Kacprzyk (Eds.), Fuzzy Logic for the Management of Uncertainty, Wiley, New York, 1992

Author Biographies

Agostino G. Bruzzone is Full Professor at the University of Genoa. He is Simulation Team MITIM DIPTEM Genoa Director for the McLeod Institute of Simulation Science (an Institution with 28 Centers distributed worldwide: Brazil, China, USA, UK, Italy, France, Germany, Canada, Spain, etc.) and he is General Director of M&S Net, an International Network involving 34 Centers. He is founding member and president of the Liophant Simulation and Simulation Team. He is also members of IEEE, IASTED, ANIMP, SCS, and MIMOS. He is active in the field of simulator-based applications for industrial plants, developing new methodologies and intelligent system integration techniques. He teaches Project Management and Industrial Logistics, "Modelling & Simulation" "M&S for Biomedical System" and VV&A for the DIMS PhD Program (Doctorship in Integrated Mathematical Modelling and Simulation). He is President of the MIPET (International Master Program in Industrial Plant Engineering & Technologies) of Genoa University. He served as an M&S Project Leader for NATO STO CMRE.

Marina Massei is currently enrolled in DIME University of Genoa as member of the Simulation Team. She is involved in the organization of International Scientific Events (i.e., I3M, WAMS, and Summer Computer Simulation Conference 2004–2006). She is Associate Director of the MISS Center, University of Perugia. She is General Cochair of MAS—The International Conference on Modelling and Applied Simulation. She served as Chair for Applications in Management, Planning and Forecasting Workshop in SummerSim. She is certified NATO CAX (Computer Assisted Exercise) operator.

Epilogue

The exploration of the phrase "simulation-based" in this book provides evidence that simulation-based thinking is prevalent in many disciplines including complex systems, engineering, medicine, natural sciences, physical sciences, social sciences, education, and training. This book provides a compendium of the state-of-the-art practices in these disciplines now supported with simulation technologies. Dedicated professionals spanning leadership, management, and engineering recognize Modeling and Simulation (M&S) as a profession and as an integral structure to their day-to-day operations.

The simulation community includes many technical societies to serve their members in many disciplines. For example, there are over 100 technical M&S societies[1] in four groups: international, national, and regional associations as well as networking of professional organizations. One of the professional associations, the Society for Modeling & Simulation International (SCS), has been in existence for over sixty years. SCS organizes Spring Simulation Multi-conferences, Summer Simulation Multi-conferences, Power Plant Simulation conferences, and the more recent, Asia Simulation Multi-conferences. We are fortunate to have members of this society include Tuncer Ören, Bernard P. Zeigler, Roy Crosbie, and Ralph Huntsinger, who collectively have been a part of SCS for over 150 years. As of the publication of this book, the Winter Simulation Conference[2] is celebrating its 50th anniversary. While these conferences are typically based in the USA, there are many more that have been active in Europe and Asia.

Simulation is a big business, with new technologies such as serious gaming, augmented reality and virtual worlds now becoming regular household entities. Furthermore, simulation used for entertainment purposes (e.g., simulation games) is also a big business. Without simulation at their core, such experiences would never have become possible.

Many professionals who use simulation techniques in their practice may not consider themselves simulationists, since they primarily identify as engineers,

[1] http://www.site.uottawa.ca/~oren/links-MS-AG.htm.
[2] http://www.wintersim.org.

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5

scientists, social scientists, or medical or defense professionals. However, the technology they are applying is simulation.

Simulation establishes a model in a computational environment and allows us to experiment with the model to solidify our understanding in a dynamic environment. "Computation" in computational modeling and simulation is an *aspect* of modeling and simulation and does not (cannot) imply that modeling and simulation is a subfield of computation or software engineering. Pretending otherwise, would be like claiming that "computational astronomy" and "computational archeology" respectively are, subfields of computation or software engineering. However, simulation extends the power of computation by allowing experimentation and experience. Furthermore, simulation models can act as generators of new data under the associated scenarios.

The advancements in simulation of cultures, human personality, and behavior as well as incorporation of social-system and ethics in simulation studies may support simulation-based rational decision-making education and training as a part of the education and training of future statesmen. This may even include education of citizens including the recognition of cognitive biases leveraged by some politicians to distort reality for their personal advantages.

Modeling and simulation practitioners are truly happy to see simulation now in every aspect of modern life. Today, we cannot (should not) embark any complex system study (design, analysis, or control) without considering "simulation-based" techniques. Simulation-based approaches in any discipline—like the proverbial sharpening the axe—is a rational way to enhance our job performance in an effective way. This book is a guide to explore the advantages of simulation-based approaches in many disciplines. By emphasizing the role of simulation in education in many disciplines, future professionals may be better equipped for their professions. Involving—even non-computational—simulation in K to 12 education, future generations may be better prepared with enhanced thinking abilities.

Saurabh Mittal,[3] Herndon, Virginia, USA
Umut Durak, Braunschweig, Germany
Tuncer Ören, Ottawa, Ontario, Canada
March 2017

[3]The authors affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved for public release: Case: 17-1658.

Author Index

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5

Subject Index

S. Mittal et al. (eds.), *Guide to Simulation-Based Disciplines*,
Simulation Foundations, Methods and Applications,
DOI 10.1007/978-3-319-61264-5