The Combined 14th Workshop for PhD Students in Object-Oriented Systems and Doctoral Symposium*

Susanne Jucknath[1], Jan Wloka[2], Eric Jul[3],
Sari R. Eldadah[4], and Ademar Aguiar[5]

[1] Technical University of Berlin, Germany
[2] Fraunhofer Institute FIRST, Berlin, Germany
[3] DIKU, University of Copenhagen, Denmark
[4] Method Building Capacity Cooperation, Jordan
[5] University of Porto (FEUP), Portugal
http://swt.cs.tu-berlin.de/ecoop04

Abstract. The PhDOOS workshop differs from other workshops because the range of participants is much smaller (only PhD students) but has a wide scope of topics. Even with the limitation to PhD students in Object Oriented Systems, the presentations covered topics such as Generic Ownership, Generic Algorithms, Model Driven Architecture, Prediction of Size, QoS assessment, Frameworks, Teaching of Frameworks and Object Calculus.

Several topics of shared interest were identified and targeted in separate discussion groups on a general theme on the future of object oriented programming. As the participants had various research interests covering very different parts of the OO spectrum, we can confidently state that these topics reflect actual concerns and needs of the OO community, and emerge from its concrete needs.

This document is to be complemented by a workshop proceedings online document which will contain the full versions of the presented papers.

1 Introduction

The 14th workshop for PhD Students in Object Oriented Systems (PhDOOS'04) was for the first time combined with a Doctoral Symposium. The combined event was held on June 14-15 2004 in Oslo, Norway in association with the 18th European Conference on Object Oriented Programming (ECOOP). The workshop was part of the series of PhDOOS workshops held in conjunction with ECOOP each year, same as the Doctoral Symposium. The decision to unite

* The title of this report sould be referenced as "Report from the ECOOP 2004 combined 14th Workshop for PhD Students in Object-Oriented Systems and Doctoral Symposium".

J. Malenfant and B.M. Østvold (Eds.): ECOOP 2004, LNCS 3344, pp. 94–100, 2004.

these events was made due to the similar group of participants. Students at the beginning of their research should attend the PhD workshop, where they can discuss their ideas with researchers at a similar stage of work and similar difficulties. Students near the finishing of their dissertation should attend the Doctoral Symposium, where they present their work in front of a group of senior researchers who will evaluate their work from a different point of view. Both – participants of the PhD workshop and the Doctoral Symposium – were free to enter the other event. The invited talk and the workshop discussion were relevant for both groups.

The participation was by invitation only, selected after a review process that included at least two reviews for each submission. Each aspirant for the Doctoral Symposium had to sent a package containing general information (name of student, name of advisor, full address, university or department), a research abstract (title of the dissertation, research area, problem to be solved, research hypothesis, approach to solve the problem) and a letter of recommendation from the students advisor. Two students have been selected to join the Doctoral Symposium (Alex Potanin and Sameh Elnikatney) and were supervised during the session by Professor Eric Jul and Professor Markku Sakkinen. The students had to give a 45 minute talk in front of them and later answer their questions for another 45 minutes.

Students, who were interested in joining the PhD-Workshop, had send a package containing general information about the authors and a paper which was limited up to 10 pages. Eight submissions were selected after the review process, so that eight students actually presented at the workshop (Po-Hao Chang, Jerome Darbon, Gregory de Fombelle, Johan Glimming, Guillermo Jimenez-Diaz, Raquel Hervas, Gabor Stikkel and Istvan Zolyomi). Each of these participants had to give a talk about her paper including a discussion part, altogether strictly limited to 45 minutes.

2 Workshop Structure

A large diversity of topics was presented, the submissions might be divided into three general groups: General Problems in Object Oriented Programming, Model Driven Architecture and Frameworks. There has also been two other works, which don't fit into this scheme. The individual topics were the following:

1. General Problems in Object Oriented Programming
   (a) Generic Ownership
   (b) C++ template Introspection Library
   (c) Generic Algorithmic Blocks dedicated to Image Processing
   (d) Design Pattern
2. Model Driven Architecture
   (a) Prediction of Size in Executable Model Driven Architecture
   (b) QoS assessment
3. Frameworks
   (a) Adaptive Distributed Object Framework for the Web
   (b) Case-Based Approach for Teaching Frameworks

4. Others
   (a) Object Calculus
   (b) Dynamic Content Delivery Network

Besides discussions of these topics Professor Eric Jul gave a talk about how to do PhD studies and how to work toward actually completing the studies with a readable PhD dissertation as a result.

The following sections summarize the contributions and discussions that took place in the various sessions.

2.1 General Problems in Object Oriented Programming

Alex Potanin presented *Practical Ownership Control in Programming Languages*, where he outlines a general problem in object-oriented languages and one solution in Java. An object is aliased whenever there is more than one pointer to that object. Aliasing can cause a range of difficult problems within object-oriented programs, because one referring object can change the state of the aliased object, implicitly affecting all the other referring objects. To deal with such problems, object instance encapsulation has been widely studied in literature. But most of the former approaches do not take the introduction of generic in modern programming languages into account. The approach here is to combine ownership and generic types and developed a compiler extension, called Ownership Generic Java (OGJ)

Istvan Zolyomi presented *A general C++ template introspection library*, mentioning Zoltan Porkolab as his advisor. Generic programming is heavily based on parametric polymorphism, what is provided by templates in C++. To ensure the correctness of template based constructions, constraints on template parameters are especially useful. Unlike other languages (Ada, Eiffel, etc), C++ does not directly support checking requirements on template parameters (concept checking), but many authors mentions ad hoc solutions based on special language features. The here presented solution is a template meta-programming library which supports an easy expression of basic concepts, providing the possibility to avoid re-implementation of simple checks for every single concept. Based on this library there is also a checking method that takes the advantages of both traditional concepts and static interfaces.

Jerome Darbon presented *Generic Algorithmic Blocks dedicated to Image Processing*, mentioning Thierry Geraud and Patrick Bellot as his advisors. The focus is here on the implementation of algorithms in the specific domain of image processing. Although many image processing libraries are available, they generally lack genericity and flexibility. Many image processing algorithms can be expressed as compositions of elementary algorithmic operations referred to as blocks. Implementing these compositions is achieved using generic programming. As a conclusion this solution is compared to previous ones and demonstrated on a class image processing algorithm.

Raquel Hervas presented *Using Design Patterns to Abstract a Software Architecture for Natural Language Generation*, joint work with Pablo Gervas. There are several problems of Natural Language Generation (NLG), including the fact that there is no consensus about the architecture a NLG must follow. The approach here is a reusable software architecture for a NLG systems by applying design patterns to abstract the main design decisions involved in the construction of a system of this kind. Its applicability had been tested through the development of a particular instantiation of a simple NLG application.

2.2 Model Driven Architecture

Gabor Stikkel presented *Prediction of Size in Executable Model Driven Architectures*, joint work with Zoltan Theisz. Most of the project planning approaches are based on the prediction of the software size. Size is often measured in lines of code. Lines of code as a size measure must be reviewed in the context of Executable Model Driven Architecture as some part of the code is generated automatically. Moreover models which are based on object-orientation are extended with code written in an action semantic language. The outcome of their investigations shows how the amount of action semantic language code can be predicted by using object-oriented design metrics. As one result the number of operations seems to be a good predictor of the lines of code written in the action semantic language.

Gregory de Fombelle presented *A model driven engineering chain for early QoS assessment in dynamic federation of systems*. Dynamic federation of systems raise the problem's quality of service properties assessment early in the development process. At the moment it is not conceivable to perform cross cutting properties evaluation on operational systems because of combinatory explosion. One should assess those properties with QoS tagged models of systems and QoS tagged models of platforms on which systems will deploy.

2.3 Frameworks

Po-Hao Chang presented *An Adaptive Distributed Object Framework for the Web*. Currently, coding Web applications is location and platform dependent: programmers entangle code for client and server together with different technologies. Due to the diversity and dynamicity of the Internet, it is hardly believed such inflexible code could deliver good performance in all the cases. The goal is to develop a framework which provide programming abstraction over the heterogeneous Internet, and the mechanism to realize the abstraction. This is composed of two tasks: a scripting language with concurrent object support and the associate retargetable compiler to provide a uniform abstraction over the Internet, and a framework management system for adaptive application deployment and resource management. The main contribution of this work is to extend the control and management of the Web out of the servers without special protocols and software.

Guillermo Jimenez-Diaz presented *A Case-Based Approach for Teaching Frameworks*, joint work with Mercedes Gomez-Allbarran. There are several learning techniques for computer skills and learning to use an object-oriented framework is a hard task. There has been little work done to develop effective techniques to reduce the effort and time taken to teach how to use a framework.

The solution here is a Case-Based Teaching approach following an active learning process where the learner is involved in resolving exercises based on framework instantiation examples.

2.4 Others

Johan Glimming presented *Difunctorial Semantics of Object Calculus*, where he gives a new denotational semantics for Abadi and Cardelli s object calculus (without sub-typing). The model is a Cartesian closed category which gives a self-application semantics of objects and model object types as fixed points of mixed variance functors. A prove is also presented that the denotational semantics agrees with Abadi and Cardelli s operational semantics. As a conclusion there are some further research directions for the denotational model explained.

Sameh Elnikety presented *DynaServer:Dynamic Content Delivery Network for E-commerce*. Traditionally a website generates dynamic web content by running a web server, application server and database server, all located in one place. The approach here is to use a network of proxies to generate and deliver dynamic web content to users. These proxies should be located at strategic points in the Internet to be close to users, achieving both low latency and high availability as perceived by uses. Each proxy maintains a complete replica of the database and should have a web server and application server that execute application code locally. To maintain data consistency among the databases replicas, the system provides generalized snapshot isolation to database transactions.

3 Workshop Discussions

A large variety of approaches were presented, spanning from foundations of object orientation over language improvements and several domain specific problems to the teaching of object oriented frameworks. In contrast to preceding PhDOOS workshops this workshop wasn't dominated by a certain new "wonder" technique, like extreme programming, refactoring or aspect-oriented programming. Single sharp bordered problems and their solutions using well known OO techniques and their improvements were in the focus of this years' PhD work. For example, problems of domains like image processing, natural language generation and web application development were presented to motivate interesting drawbacks of object oriented means.

In several discussions many different OO techniques were individually depicted and their problems and applications discussed in small groups. Some features of C++, framework based solutions and generative programming built the core of presented PhD work. The improvement of OO programming language or the creation of a specific framework seems to help usual developers to keep focussed to their actual problem solution.

Concluding Discussion Participants: Jerome Darbon, Jan Wloka, Susanne Jucknath, Po-Hao Chang, Raquel Hervas, Guillermo Jimenez-Diaz, Sameh Elnikety

In a concluding discussion we tried to identify next hot topics in object orientated research. To start with a tangible base, requirements for descriptions of certain computation in different domains were identified. For example, the composition of functional units seems to be an important capability in image processing whereas a transparent distribution of objects is essentially needed for the development of web applications. Developers struggle in every domain with different abstractions. At first most participants agreed in the need for more flexibility in programming languages. However, a small excursion to the obstacles in teaching frameworks gave enough suitable examples for difficulties in handling flexibility. Recognizing the fact that the lost of comprehension and control is the effect caused by to much flexibility, the wish of bit more ”on demand” adaptation was generated. A kind of adaptation mechanism that could be easily configured to the specific needs of a certain domain.

After many and also partially turbulent discussions a lot of different views, ideas and techniques have been exchanged. As a common agreement could be seen that developers need to describe their application at various abstraction levels. On the one hand there are non-computer scientists, like mathematicians or engineers, writing programs focussed on algorithms that solve abstract computation problems. There is usually no specific need to describe the underlying implementation details. On the other hand in high-constraint environments, where a certain performance in terms of response time or physical data cache is important, one can be interested in very low level implementation details. Hence, developers are still eager for a mean that combines simplified abstraction and ”on-demand” flexibility in descriptions of computation.

Acknowledgments

We would like to thank AITO for financial support, which was important for many students. AITO provided support to partially cover student travel expenses. As we had participants from Sweden, France, Spain, Germany, Hungary, USA, and New Zealand—and most of them were not financially provided for as they are students—we were most grateful for support.

Participants List

1. Ademar Aguiar, University of Porto, Portugal
   ademar.aguiar@fe.up.pt
2. Patrick Bellot, ENST - INFRES, France
   bellot@enst.fr
3. Po-Hao Chang, University of Illinois at Urbana-Champaign, USA
   pchang2@uiuc.edu
4. Jerome Darbon, EPITA - LRDE, France
   darbon@lrde.epita.fr, darbon@enst.fr
5. Sari R. Eldadah, Method Building Capacity Cooperation, Jordan
   Sdadah@MethodCorp.com

6. Sameh Elnikety, School of Computer and Communication Sciences, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
   sameh.elnikety@epfl.ch
7. Gregory de Fombelle, Thales Research and Technology, France
   gregory.defombelle@thalesgroup.com
8. Thierry Geraud, EPITA - LRDE, France
   geraud@lrde.epita.fr
9. Pablo Gervas, Universidad Complutense de Madrid, Spain
   pgervas@sip.ucm.es
10. Johan Glimming, Stockholm University, Sweden
    glimming@kth.se
11. Mercedes Gomez-Albarran, Universidad Complutense de Madrid, Spain
    albarran@sip.ucii.es
12. Raquel Hervas, Universidad Complutense de Madrid, Spain
    pgervas@sip.ucm.es
13. Guillermo Jimenez-Diaz, Universidad Complutense de Madrid, Spain
    gjimenez@sip.ucii.es
14. Susanne Jucknath, Technical University of Berlin, Germany
    susannej@cs.tu-berlin.de
15. Eric Jul, DIKU, University of Copenhagen, Denmark
    eric@diku.dk
16. Zoltan Porkolab, Eoetvoes Lorand University, Hungary
17. Alex Potanin, Victoria University of Wellington, New Zealand
    alex@mcs.vuw.ac.nz
18. Gabor Stikkel, Lorand Eoetvoes University, Hungary
    stiko@compalg.inf.elte.hu
19. Zoltan Theisz, Ericsson Telecommunications Hungary
    gmd@alta.hu
20. Jan Wloka, Fraunhofer Institute FIRST, Germany
    Jan.Wloka@first.fraunhofer.de
21. Istvan Zolyomi, Eoetvoes Lorand University, Hungary
    scamel@alta.hu