Adaptability in Object-Oriented Software Development Workshop report

Bedir Tekinerdogan & Mehmet Aksit
Department of Computer Science
University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
E-Mail: bedir@cs.utwente.nl

Abstract

This paper is a report of the ECOOP '96 adaptability in object-oriented software development workshop held on July, 8th, 1996 in Linz. The basic topics raised during the workshop are summarized. Conclusions and a list of attendees are given as well.

1 Introduction

The workshop was organized by the University of Twente, Enschede, The Netherlands (Mehmet Aksit, Bedir Tekinerdogan & Lodewijk Bergmans) in coordination with the Free University of Brussels, Belgium (Kim Mens, Patrick Steyaert & Carine Lucas) and the Northeastern University, Boston, US (Karl Lieberherr).

Large and complex software systems have to cope with a steady change of various requirements. The reasons for this are twofold. First, complex systems are very hard to understand and therefore the initial requirements of these systems suffer from errors. Second, complex systems tend to evolve after they have been developed and consequently the requirements change accordingly. In order to cope with these problems the demand has risen for adaptable software which can be adjusted to the changing requirements or correct the errors in the initial requirements engineering.

Adaptability can be defined at various levels e.g. analysis and design level, program level, or compilation and link-time level. Several research activities have paid attention to reusability and adaptability. There are however still a number of open research issues in designing adaptable software systems which need to be solved.

This workshop tried to exploit the problems of adaptability models and techniques and explore new generic mechanisms to enhance the adaptability at various abstraction levels.

The workshop received 12 position papers and was attended by 31 participants from 11 different countries. A number of selected participants presented position papers after which more interactive discussions were held. To facilitate lively discussions and the exchange of ideas the presentations were hold short (5 minutes per presentation). The agenda is given in the following section.

2 Agenda

9:00-9:15 Opening and Introduction by Mehmet Aksit

9:15-9:45 Session: **THEORY**, general adaptability issues & characteristics

- Mehmet Aksit (5’): *Separation and Compositon of Concerns*.
- Karl Lieberherr (5’): *From Transience to Persistence in OO Programming: Aritectures and Patterns.*
- Bedir Tekinerdogan (5’): *Modeling Adaptability in OO software Development*
- General Discussion (15')

9:45-10:15 Session: **FRAMEWORKS**

- Kim Mens, Patrick Steyaert, Carine Lucas (5’): *Reuse Contracts: Managing Evolution in Adaptable Systems*.
- Benoît Garbinato, Pascal Felber, Rachid Guerraoui (5’): *Right Abstractions on Adequate Frameworks for Building Adaptable Distributed Applications*.
- Krzysztof Czarnecki (5’): *Concreteness kills Adaptability*.
- General discussion (15’)

10:15-10:30 BREAK

10:30-11:00 Session: **DISTRIBUTED SYSTEMS**

- Frank Schubert (5’): *Dynamic Adaptability in Operating Systems by Means of an Object-Oriented Architecture*.
- Tuomas Ihme & Eila Niemelä (5’): *Adaptability in Object-Oriented Embedded and Distributed Software*.
- General discussion (15’)

11:00-11:30 Session: **SPECIAL TOPICS**

- Lodewijk Bergmans (5’): *Composability and Adaptability*.
- Anders Mørch (5’): *Adaptation through End-user Tailoring*.
- Naftaly H. Minsky & Partha pratim Pal (5’): *Multiple Views for Objects as a Means for Adaptation*.
- Patrick Lefebvre & Michel de Heaulme (5’): *Information Systems Require a Re-Examination of Analysis and Design Criteria*.
- General discussion (15’)

11:30-12:30 Group splitting

Split the group in sub-groups which will discuss specific topics in the afternoon.

12:30-14:00 Lunch break

14:00-16:00 Session: Sub-Group discussion

1. Cost and efficiency
2. Adaptability Constraints
3. Separation of Concerns

16:00-17:30 Group Reports & Conclusions and evaluations.

3 Morning Sessions

The morning session started with an introduction by Mehmet Aksit. He made the observation that we have to deal with two contradictory goals. On the one hand, we need to *fix* the design artifacts and software components for robustness. On the other hand, we need to make the design artifacts and software components *variable* for maintainability. Clearly, some compromise between both goals is required. The submitted papers can actually be classified in two ways:

1. On the exploition of the trade-off between robustness and adaptability at the specification and implementation level.
2. On the ways how different concerns are integrated in one system given the level of adaptability needed.

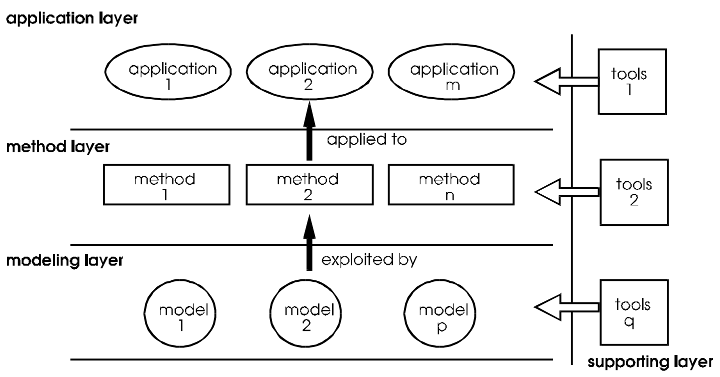

Figure 1. Architecture for software development

Figure 1 presents an object-oriented software development framework. This architecture consists of four basic layers. The top layer is the application layer and represents the application being developed using this architecture. The method layer includes process descriptions, notations, rules and hints to build the application with the existing computation models. The computation models represent the basic building blocks of the application and include the object-oriented features like objects, classes, messages and inheritance. The supporting layer provides tools to support the horizontal layers, like dedicated compilers and

CASE tools. Although, adaptability is generally needed at all these layers, most studies on adaptability focus only on a single layer

There are two approaches for adaptation, transformational adaptation or compositional adaptation. In the first case the system or part of the system is adapted by transforming a specification and/or implementation in such a way that it fulfills the changing requirements. In the latter case, adaptation takes place using compositional techniques, that is, by composing different subcomponents to provide software offers the required functionality.

The presentations and the discussions of the position papers were classified in 4 sessions. The first session was related with the theory and general adaptability issues & characteristics. In the second session object-oriented application frameworks were discussed as one of the basic concepts for developing adaptable software. The third session handled about adaptability in distributed systems. Finally, in the last session more specific topics in relation with adaptability were discussed, e.g. impact of composability on adaptability, specific techniques to improve adaptability, end-user tailoring etc.

4 Group Discussions

During the afternoon, participants were split into small subgroups to discuss the main issues which were raised during the presentations. The classification of the topics together with the issues raised were as follows:

1. Separation of Concerns

It is commonly agreed that the separation of concerns principle will support the adaptability. The separation of concerns principle enforces the separation of different aspects of a system both at the conceptual as well as on the implementation level. This will increase the understandability, the reusability and the maintainability of the single concerns. Together, this will facilitate the integration of the different concerns.

2. Adaptability Constraints

Adaptation should be subject to and guided by a set of constraints. Systems and constraints will evolve in parallel. The question is then how to deal and regulate changes in constraints. In order to effectively deal with these changing constraints it is important to identify and distinguish the useful and enforceable constraints. Finally, the constraints may be enforced at different layers of the software development framework in figure 1.

3. Cost and efficiency

The costs involved for obtaining adaptability occur at various phases in a product’s lifetime. During product development, organization restructuring, programmer training and the initial cost of rewriting a system have to be taken into account. In the operational phase, execution speed and memory consumption are the issues, which may be subsumed under the notion of efficiency. Manufacturers save software rewriting costs when requirements or hardware changes can increase their market share by widening their product's range of applicability. Versatility and performance may be increased by dynamically adapting to changing runtime conditions. There is no common answer to the question whether the gains justify the costs, ‘‘It depends”. Just as in any other area of life, the amortized gain of an action should be positive in the long run.

4 Selected Papers

The following papers were eventually submitted and selected as publication in this book:

- Mehmet Aksit, Bedir Tekinerdogan, & Lodewijk Bergmans: *Achieving Adaptability through Separation and Composition of Concerns*.
- Benoit Garbinato, Pascal Felber, & Rachid Guerraoui: *Right Abstractions on Adequate Frameworks for Building Adaptable Distributed Applications*.
- Tuomas Ihme & Eila Niemelä: *Adaptability in Object-Oriented Embedded and Distributed Software*.
- Kim Mens, Patrick Steyaert & Carine Lucas: *Reuse Contracts: Managing Evolution in Adaptable Systems*.
- Anders Mørch: *Adaptation through End-user tailoring*.

5 Conclusions

Some of the basic conclusions and issues which were discussed could be summarized as follows:

- Adaptability is defined as the ease with which a system or parts of the system may be adapted to the changing requirements.
- Adaptation can take place at various levels e.g. analysis and design level, program level, or compilation and link-time level.
- We have to cope with two contradictory goals. On the one hand we would like to have fixed software components for robustness and stability. On the other hand we need variable software artifacts for adaptability and maintenance. The basic question is how to find a compromise between robustness and adaptability.
- There are two approaches for adaptation, transformational adaptation or compositional adaptation. In the first case the system or part of the system is adapted by transforming software artifacts in such a way that they fulfill the requirements. In the latter case, adaptation takes place using compositional techniques, that is, by composing different components to provide software components which offer more complex or more adjusted functionality.

- Composability facilitates the construction of a software system and makes systems easier to adapt. Adaptability facilitates the response to changing requirements and is therefore related with the evolution of the system. Further, adaptability makes building blocks easier to compose.
- The conventional design processes are problematic because they are based on subsequent elimination of alternatives and early commitments which reduce the adaptability.

For more information about this workshop, including a list of participants, position papers check out the website at:

http://wwwtrese.cs.utwente.nl/ecoop96adws/.

6 List of attendees

For each participant the name, affiliation and e-mail address is listed:

Mehmet Aksit - University of Twente - aksit@cs.utwente.nl
Joaquim Baptista - Univ. Nova de Baptista - px@fct.unl.pt
Lodewijk Bergmans - University of Twente - bergmans@cs.utwente.nl
Rainer Blome - Univ. of Göttingen - rainer@physik3.gwdg.de
Pim van den Broek - University of Twente - pimvdb@cs.utwente.nl
Krzystof Czarnecki - Daimler-Benz Research - czarnecki@dbag.ulm.DaimlerBenz.com
Pierre Cointe - Ecole des Mines de Nantes - cointe@emn.fr
Benoît Garbinato - EPFL/Lausanne - garbinato@lsesun2.epfl.ch
Rachid Guerraoui - EPFL -rachid@lse.epfl.ch
Juan Hernandez - University of Extremadura - juanher@unex.es
Theo D'Hondt - Free University of Brussels - tjdhondt@vub.ac.be
Tuomas Ihme - VTT Electronics - tuomas.ihme@vtt.fi
Gregor Kiczales - XEROX PARC - gregor@parc.xerox.com
Aleksander Klosow - Techn. Univ. of Wroclaw - klosow@ictadmin.ict.pwr.wroc.pl
Piet S. Koopmans - University of Twente - koopmans@cs.utwente.nl
Thomas Kotzmann - University of Linz -
Patrick Lefebvre - AIM Pitie-Salpetriere - plefebvre@pratique.fr
Karl Lieberherr - Northeastern University - lieber@ccs.neu.edu
Carine Lucas - Free University of Brussels - clucas@vnet3.vub.ac.be
Theo Dirk Meijler - University of Bern - meijler@iam.unibe.ch
Naftaly Minsky - Rutgers University - minsky@cs.rutgers.edu
Anders Morch - Univ. of Oslo - andersm@ifi.uio.no
John Mordhorst - Clemson University - mordhorst@cs.clemson.edu
Oscar Nierstrasz - University of Berne - oscar@iam.unibe.ch
Johan van Oeyen - University of Leuven - Johan.VanOeyen@cs.kuleuven.ac.be
Fernando Sanchez - University of Exttremadura - fernando@unex.es
Frank Schubert - Techn. Univ. Chemnitz - fsc@informatik.tu-chemnitz.de
Patrick Steyaert - Free University of Brussels - prsteyae@vub.ac.be
Gunnar Teege - Technische Universität München - teege@informatik.tu-muenchen.de
Bedir Tekinerdogan - University of Twente - bedir@cs.utwente.nl

Acknowledgements

The organizers would like to thank all the participants of the workshop.