T U M

I N S T I T U T F Ü R I N F O R M A T I K

RTSE'97 - Workshop on Requirements Targeting Software Engineering

Manfred Broy, Bernhard Rumpe

TUM–I9807
Dezember 98

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-12-I9807-100/1.-FI
Alle Rechte vorbehalten
Nachdruck auch auszugsweise verboten

©1998

Druck: Institut für Informatik der
Technischen Universität München

Foreword

In Europe and North America the competence in software engineering research has different profiles. While in North America there is a lot of know how in the practical, technical, and organisational aspects of software engineering, in Europe the work concentrates more on foundations and formal modelling of software engineering issues. Both approaches have different strengths and weaknesses. Solely practice driven research in software engineering is in the danger of developing into a shallow field and could fail to find a solid scientific basis and contribute substantially to the progress in software engineering. Work concentrated on formal aspects only is in the danger of becoming too theoretical and isolated from practice such that any transfer into practical application will fail.

Substantial progress in software engineering can be achieved by bringing together pragmatic and foundational work in software engineering research. This can provide a step towards a more common scientific basis for software engineering that allows us to integrate the various results of research and workshops, leading to fruitful synergetic effects. It will also help to identify critical research paths and developing an adequate paradigm for the scientific discipline of software engineering.

It was the goal of this workshop to bring together experts from science and practice in software and systems engineering from North America and Europe.

In software and systems engineering it is necessary to distinguish the enormous difference between the dynamics in development we refer to and the limited scope assumed by many of today's software managers that still use outdated techniques. Many of the unsolved problems associated with the old techniques are symptoms of lack of formalization and lack of automation support.

The intended focus of the workshop was on unified sets of formal models and associated methods suitable for automation for many aspects of software development, in particular those that address change and those that apply on a large scale. Some of the intended aspects of software evolution are

- modifiable software architectures,
- resource changes,
- context changes,
- requirements changes,
- changes to decomposition structures, and
- changes in plans.

These issues are related to formal representations of the version history, and formal representations of the activities that produced existing versions or have been proposed to produce future versions.

The essence of the problem is to establish and maintain consistency among various kinds of software artefacts throughout the development and evolution process, including consistency between requirements, architectures, and programs. Automation support is needed to determine dependencies and to use this dependency information to provide decision aid for software synthesis, analysis, and evolution. Many versions of each artefact are produced as the software evolves, and changes to the dependency structure must be recognised and reacted to. The challenge is to better formalise the problems in this area, and to develop some of the badly needed technical solutions.

If we as a community can succeed in this, the results will provide convincing evidence that formal methods can have strong practical value, and help reverse the trend of weakening support for the subject from both industry and governments. It seems that previous work on formal methods can be applied to problems related to these topics, but it may require non-traditional approaches. The challenge helped to trigger new ideas at the workshop, and perhaps opened new opportunities for progress.

It is well recognised in the meanwhile that software and systems engineering as an important issue in technical systems still lack a proper scientific basis. The many efforts in academia, especially under the heading formal methods, towards such a scientific basis have produced many valuable and interesting scientific results; however, most of the work of integrating this with the practice of software engineering is still missing. Nevertheless, we can observe a starting trend to bring together practical considerations and approaches with scientific results. A good example is the Unified Modelling Language that recently was designed and still will evolve. The fact that a proper semantic basis is needed for a proper methodological support is much more recognised than in its predecessors. Nevertheless, more efforts are necessary to give the scientific research more focus w.r.t. the questions that are important for practice and to stimulate a transfer between academia and application. It was the goal of the workshop to contribute to this task.

The workshop took place in early October 1997 in Bernried in Germany. It fulfilled the expectations formulated above. It is our pleasure to thank Sascha Molterer for his excellent help in organising the workshop and the Army Research Office and in particular Dave Hislop for the generous financial support.

March 1998 Manfred Broy, Bernhard Rumpe

Table of Contents

Foreword 3

A Discipline for Handling Feature Interaction 7
Egidio Astesiano, Gianna Reggio
(Dipartimento di Informatica e Scienze dell'Informazione, Genova, Italy)

Merging Changes to Software Specifications 29
V. Berzins
(Naval Postgraduate School, Monterey, CA, USA)

Domains as a Prerequesite for Requirements and Software – 39
Domain Perspectives & Facets, Requirements Aspects and Software Views
Dines Bjørner

Rapid Prototyping and Incremental Evolution 103
(abstract)
David A. Dampier
(National Defense University, Fort Lesley J. McNair, DC, USA)

Combining and Distributing Hierarchical Systems 105
Chris George
(United Nations University, Macau)
Đỗ Tiến Dũng
(Ministry of Finance, Hanoi, Vietnam)

Software Engineering Issues for Network Computing 123
Carlo Ghezzi, Giovanni Vigna
(Politechnico di Milano, Milano, Italy)

A Two-Layered Approach to Support Systematic Software Development 137
Maritta Heisel
(Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany)
Stefan Jähnichen
(Technische Universität Berlin, and also: GMD FIRST, Berlin, Germany)

Requirements Engineering Repositories: Formal Support for Informal 157
Teamwork Methods
Hans W. Nissen, Matthias Jarke
(RWTH Aachen, Aachen, Germany)

Formal Models and Prototyping 183
Luqi
(Naval Postgraduate School, Monterey, CA, USA)

Deductive-Algorithmic Verification of Reactive Systems 195
(extended abstract)
Zohar Manna, Nikolaj S. Bjørner, Anca Browne, Michael Colón,
Bernd Finkbeiner, Mark Pichora, Henny B. Sipma, Tomás E. Uribe
(Stanford University, Stanford, CA, USA)

Reference Architectures and Conformance 203
Sigurd Meldal
(Stanford University, Stanford, CA, USA)

Fine-grained and Structure-oriented Document Integration Tools are 229
needed for Development Processes
S. Gruner, M. Nagl, A. Schürr
(Aachen University of Technology, Aachen, Germany)

Software and System Modeling Based on a Unified Formal Semantics 245
M. Broy, F. Huber, B. Paech, B. Rumpe, K. Spies
(Technische Universität München, Munich, Germany)

Formal Methods and Industrial-Strength Computer Networks 269
J. N. Reed
(Oxford Brookes University, Oxford, UK)

A Framework for Evaluating System and Software Requirements 281
Specification Approaches
Erik Kamsties, H. Dieter Rombach
(Fraunhofer Institute for Experimental Software Engineering,
Kaiserslautern, Germany)

Postmodern Software Design with NYAM: Not Yet Another Method 297
Roel Wieringa
(Vrije Universiteit, Amsterdam, the Netherlands)

Index of Authors 321