Lecture Notes in Computer Science 1169
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Advisory Board: W. Brauer D. Gries J. Stoer

Manfred Broy Stephan Merz
Katharina Spies (Eds.)

Formal Systems Specification

The RPC-Memory Specification Case Study

Springer

Volume Editors

Manfred Broy
Stephan Merz
Katharina Spies
Technische Universität München, Institut für Informatik
Arcisstr. 21, D-80290 München, Germany
E-mail: {broy/merz/spies}@informatik.tu-muenchen.de

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Formal systems specification : the RPC-memory specification case study / Manfred Broy ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1996
(Lecture notes in computer science ; Vol. 1169)
ISBN 3-540-61984-4
NE: Broy, Manfred [Hrsg.]; GT

CR Subject Classification (1991): C.2.4, D.2.1, D.2.4, D.2.10,F.3.1

ISSN 0302-9743
ISBN 3-540-61984-4 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10549153 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Preface

The growing interest in a scientifically based method for the specification, refinement, implementation, and verification of distributed, concurrent, interacting systems is caused by the wide areas of their applications. These include operating systems, computer networks, telecommunications, wide-area nets, such as the Internet or the World Wide Web, and all kinds of reactive systems embedded in technical devices such as in the automotive and avionics industries. Other typical applications are business applications and hardware/software codesign. New applications are emerging rapidly. As it turned out in the 1970's, providing good methods, techniques, models, and notations for system development is not a trivial task at all. Therefore the area of the formal foundation of the development of concurrent interacting systems has attracted many researchers since the stimulating work by Edsger Dijkstra, Tony Hoare, and others.

The subject of distributed interactive systems also leads to intricate theoretical problems, which have stimulating relationships to many areas of mathematics, logics, and electrical engineering. Not surprisingly, some of the work in these areas became more and more focused on isolated problems rather separated from the original motivation, namely to provide a scientifically well-founded and practically applicable development method that includes specification, refinement, and verification. A second problem is that over the years a large and confusing variety of different approaches have been developed. This richness, reflecting the large amount of creativity and new ideas of researchers, provides us with an overwhelming variety of aspects for the design and the analysis of systems. On one hand, this is good, since it gives a richer source of material on which practical techniques could be based. Unfortunately, this can be confusing for practical engineers who are interested in improving their methods by providing formal foundations. Even for the more research-oriented engineers in that area, it is not easy to keep track of all the different approaches and to understand their advantages and disadvantages.

Case studies can provide a basis for comparing, relating, and evaluating various approaches. There have been several actions in the past to provide case studies and to compare different methods on the basis of these case studies. Some of them provided a variety of different cases for the application. As a result, for every case study only a small number of approaches were applied. Therefore the comparison between a large number of methods was not supported. In another experiment, the case study was rather large, so not too many solutions were presented due to the overhead needed to work out the case study.

But even in these cases, it became clear that case studies provide a very valuable opportunity to improve our possibilities to compare and further develop our methods. First of all, they provide a real basis to evaluate approaches. And, more than the examples presented by the various methods by themselves, they really test how far a method is able to adapt to particular requirements and methodological needs. More important, there is another remarkable effect. The test cases guide the attention of the researchers in the applicability of their

methods. As a result of several case studies, researchers had to admit that in their current form their methods were actually not applicable and were forced to extend and modify their theories, notations, and techniques. This way, an interesting feedback was set up between the attempt to provide a solution to a case study and the further development of the methods. Obviously, case studies help to focus the attention of the researchers onto the critical issues in the development task.

Based on these considerations I suggested to Leslie Lamport in fall 1993 that we again do an experiment by providing a further case study. For a fair comparison, of course, case studies should be sufficiently large to meet the actual requirements of practical applications. Moreover a variety of case studies is needed since the different approaches may be better suited for different areas of applications and therefore a comparison on the basis of only one specific case study is certainly not fair. After several meetings and discussions, Leslie and I decided to provide just one case study and to choose its size to be not too large, however, to make sure that several steps of development were part of the exercise. The reason was that we were interested in looking not only at the pure specification task but also at refinement and verification.

Finally, we decided to choose an example that is typical for the area of operating systems and hardware applications, at least on the logical level of hardware architecture, and to include in this example a number of typically important issues such as failures, fairness, and real time. Leslie wrote a first version of the problem description and, after a few iterations, the problem statement was sent out together with the invitations to a workshop in Dagstuhl.

The workshop, which took place in September 1994, was very stimulating. Originally, we were afraid that after two days the participants would be completely fed up with the problem. As it turned out, this was not the case at all. On the contrary, new aspects of the problem were discovered and discussed. Striking was the fact that for many of the presented solutions, the ideas on modeling the problem and solution proved more important than the chosen methods and notations. Correspondingly, the individual methods were often just used as a representation vehicle and did not govern the basic specification, refinement, and verification ideas. This showed to a large extent that the notation and theoretical basis should not be over-emphasized and that the methods and techniques to find good ideas for modeling, representing, and refining systems should also attract enough attention.

After the workshop, we went through several iterations to collect written versions of the presented solutions. We went through two stages of a careful cross-refereeing process and finally came to the text presented here.

After many discussions on the issue of providing a careful comparison and evaluation of the written solutions we very modestly decided to include only a short synopsis paper in this volume in which the approaches were roughly explained and related. Clearly much more work can and needs to be done here. However, being involved in some of the solutions ourselves, it would not have been fair for us to draw too far-reaching conclusions. So we leave it to the reader

and to future activities to provide a more careful analysis and comparison of the results of our experiment.

Looking back at the whole activity, I consider this experiment a success and a helpful contribution to the research community as well as to the engineering community for those interested in experimenting with formally based approaches to the design of systems.

It is a pleasure to thank the people who helped put this experiment forward. My thanks go to Leslie Lamport for an effective and stimulating cooperation, to Schloss Dagstuhl for providing an excellent place for the workshop, to Stephan Merz who helped a lot in organizing the workshop and carefully read the contributions in the refereeing process, and to Katharina Spies who helped in many ways in correspondence with the authors and who showed an admirable patience with all the little problems and difficulties that arose in the refereeing process. Finally my thanks go to Springer-Verlag who helped in a very gentle and efficient way to publish this experiment as a volume in the Lecture Notes in Computer Science.

Munich, July 1996 Manfred Broy

Acknowledgement

There is no book without authors. This is especially true for this collection of articles since we misused authors as reviewers for the contributions. We did this move since we considered the authors of articles as the best experts of the specified problem. Moreover, we were interested in an interaction between the proponents of the quite different methods. For all these reasons, we decided to establish a cross-refereeing process.

It is our pleasure to thank all the reviewers

M. Abadi,
E. Astesiano,
D. Barnard,
E. Best,
J. Cuéllar,
R. Gotzhein,
J. Hooman,
H. Hungar,
B. Jonsson,
J.N. Kok,
R. Kurki-Suonio,
L. Lamport,
K. Larsen,
S. Merz,
O. Müller,
M. Nielsen,
G. Reggio,
M. Rinderspacher,
B. Steffen,
K. Stølen,
K. Sunesen,
C. Weise

for their prompt and precise work in the last few months. Only because of their readiness for fruitful and mostly peaceful discussions and cooperation with the authors we can present contributions of selected quality in this collection. We are especially grateful for the cross-refereeing efforts to prepare assessments of the articles which were very helpful in the preparation of the synopsis you find in this volume.

Table of Contents