Lecture Notes in Computer Science 1322
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

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

Heinrich Hußmann

Formal Foundations for Software Engineering Methods

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany

Juris Hartmanis, Cornell University, NY, USA

Jan van Leeuwen, Utrecht University, The Netherlands

Author

Heinrich Hußmann
Siemens AG, Public Communication Networks, Advanced Development
Hofmannstraße 51, D-81359 Munich, Germany
E-mail: hussmannh@acm.org

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Hußmann, Heinrich:
Formal foundations for software engineering methods / Heinrich Hußmann. - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1997
(Lecture notes in computer science ; 1322)
ISBN 3-540-63613-7

CR Subject Classification (1991): D.2

ISSN 0302-9743
ISBN 3-540-63613-7 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law

© Springer-Verlag Berlin Heidelberg 1997
Printed in Germany

Typesetting: Camera-ready by author
SPIN 10545751 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Table of Contents

Chapter 1: Introduction 1
1.1 Two Worlds of Software Engineering 2
1.2 Requirements Specification for Information Systems 6
1.3 Synergy between the Pragmatic and the Formal World 8
1.4 Overview 11

Chapter 2: Terminology 13
2.1 Development Methods 13
2.2 Formality 15
2.3 Pragmatism 17

Chapter 3: SSADM - A Pragmatic Requirements Engineering Method 19
3.1 Motivation for the Choice of SSADM 19
3.2 Modelling Techniques and Methodical Support in SSADM 21
3.3 Hotel Agency: A Toy Project in SSADM 25

Chapter 4: SPECTRUM - A Formal Specification Framework 27
4.1 Characteristics of Spectrum 27
4.2 Axiomatic Specifications in Spectrum 28

Chapter 5: Alternative Ways to an Integration of SSADM and SPECTRUM 37
5.1 Contributions of SSADM to an Integrated Approach 37
5.2 Contributions of Spectrum to an Integrated Approach 44
5.3 Options for the Integration 47
5.4 Comparison with Other Approaches 58

Chapter 6: A Semantic Reference Specification for SSADM 61
6.1 Architecture of the Formal Basis for SSADM 61
6.2 Core Reference Specification 66
6.3 Core Extensions for Logical Data Modelling 69
6.4 Core Extensions for Entity-Event Modelling 71

Chapter 7: The SSADM Technique "Logical Data Modelling" 75
7.1 Entity-Relationship Diagrams 75
7.2 Formalization of ER Diagrams 80
7.3 Entity Descriptions and Attributes 88
7.4 Formalization of Entity Descriptions 90
7.5 LDM Project Level Translation Schemes 95

Chapter 8: The SSADM Technique "Entity-Event Modelling" 99
8.1 EEM: Introduction and Terminology 99
8.2 Event-Entity Matrix 104
8.3 Effect Correspondence Diagrams 117
8.4 Entity Life Histories 135

Foreword

Formal methods in computer science aim at a scientific foundation on which methods and tools for practical system construction can be based. Practice requires methods for the specification of software systems, tools for checking specifications for syntactic consistency, and support for generation of code from specifications, as well as tools for proving the correctness of manually produced code with respect to the specification. Unfortunately, the practical relevance of many publications on formal methods is limited.

On the other hand, many methods and tools used in practice nowadays do not make use of the current status in computer science research. They are lacking clear concepts, not to speak of a precise definition based on formal techniques.

Therefore, there is a need for research aiming at methods that are usable in practice and theoretically founded, in order to provide a basis for powerful tools in software engineering.

In this book Hussmann builds a bridge between the "pragmatic" methods for design of information systems and the formal mathematical background, by giving a formal definition for the method SSADM within the algebraic specification language SPECTRUM. SSADM, a British standard method belonging to the so-called structured methods, serves here as an example. Although SSADM has its methodical deficiencies, the example is well chosen since it is practically relevant. Moreover, it is well suited to demonstrate the power of formal techniques since it is sufficiently complex.

The book gives a complete formal definition for SSADM. This shows that such an approach is not only a theoretical possibility but is also feasible for methods of practical relevance. This is made possible by the usage of clear structuring concepts to build up the formal definition.

The formal foundation of SSADM is more than an example. It is presented in a way that can be seen as a general pattern according to which other software engineering methods also can be defined, in particular the modern object-oriented methods.

Our wish for this book is that it encourages other scientists to apply theoretical foundations of computer science to practical problems of software development in a similarly thorough way. Thus the importance of this book goes beyond SSADM and SPECTRUM.

Munich, November 1996

Ernst Denert
Manfred Broy

Preface

Research in software engineering proceeds in two almost unrelated streams of results. The *formal* stream is an offspring of mathematical logic and relies completely on the usage of a symbolic language. The *pragmatic* stream is derived from decades of practical development of large systems and applies informal (or only weakly formal) notations, like diagrams, tables and natural language.

This book contributes to bridging the gap between the formal and the pragmatic approaches to the systematic construction and description of software. The central claim of this book is that a thorough integration of formal and pragmatic methods is possible, and also fruitful, since it leads to significant improvements for both areas.

Formal and pragmatic methods are to some extent complementary, so it is possible to compensate the deficiencies of one approach with the other. This is particularly valid for the early phases of software development where requirements are developed and specified. Therefore, this book concentrates on the area of requirements engineering. Moreover, the focus is on methods for the development of business information systems. As a representative for a practically relevant method for the definition of requirements for information systems, the British standard method SSADM is chosen. A formal semantics for the key concepts of SSADM is given, using the formal specification language SPECTRUM.

This work presents results of three different kinds. First, the principal feasibility of an integration between formal and pragmatic methods is demonstrated. Second, the formalisation is used as a thorough semantic analysis of the concepts in SSADM. Third, a way is shown to obtain a hybrid formal-pragmatic specification using a mixture of SSADM notations and formal (SPECTRUM) specifications.

This book is based on a Habilitation Thesis which was accepted by the Technical University of Munich in spring of 1995. I would like to thank Professor Manfred Broy for his intensive support and encouragement for this work. I am grateful for the opportunities to exchange opinions with people of different background, including experienced practicians like Professor Ernst Denert as well as many highly motivated students. I gratefully acknowledge the excellent atmosphere among my collegues at the Technische Universität München. In particular I would like to thank Bernhard Rumpe and Cornel Klein for reading draft versions of this thesis, and Herbert Ehler, Rudolf Hettler and Friederike Nickl for important suggestions.

Last but not least I would like to thank my wife Gabi, with excuses for having spoiled so many weekends and evenings with work.

Tutzing, January 1997 Heinrich Hussmann

Notice

This work is based on the software development method SSADM, version 4. The rights for SSADM are held by the British CCTA. This work does not claim to be compliant with the original SSADM method, and any statements made about SSADM are the personal opinion of the author.

This work has been partially supported by the Bayerische Forschungsstiftung through a cooperation of the Technische Universität München with FAST e.V. (Forschungsinstitut für angewandte Software-Technologie).

8.5 Operation Lists 146
8.6 EEM: The Whole Picture 153
8.7 EEM Project Level Translation Schemes 159

Chapter 9: Logical Consistency of SSADM-F 161
9.1 Proof of Consistency by Executable Implementation 162
9.2 An Executable Specification for SSADM–F 164
9.3 Logical Prototyping of SSADM-F Specifications 180

Chapter 10: Formal Extensions for SSADM-F 185
10.1 Combining Pragmatic and Formal Specifications 185
10.2 Extensions for Entity-Event Modelling 188
10.3 Interfaces between Spectrum and SSADM-F 194
10.4 Spectrum Extensions for SSADM-F 198
10.5 Perspectives for Formally Extended Pragmatic Specifications 205

Chapter 11: The SSADM Technique "Data Flow Modelling" 207
11.1 Data Flow Diagrams in SSADM 207
11.2 Formalization of Data Flow Diagrams 210
11.3 Integration between DFM and LDM 215
11.4 Integration between DFM and EEM 218

Chapter 12: Summary and Outlook 225
12.1 Summary 225
12.2 Evaluation 226
12.3 Outlook 229

Appendix A: Glossary 233

Appendix B: "Hotel Agency" - An SSADM-F Specification 241
B.1 DFD Documents 242
B.2 LDM Documents 242
B.3 EEM Documents 245

Appendix C: SPECTRUM Translation of the "Hotel Agency" Specification 253
C.1 Translation of LDM Documents 253
C.2 Translation of EEM Documents 256

Appendix D: Basic SPECTRUM Specifications 265
D.1 Lists 265
D.2 Finite Sets 267
D.3 Finite Maps 268

Bibliography 271

Subject Index 279