

Lecture Notes in Computer Science

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

1646

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Singapore

Tokyo

Bernhard Westfechtel

Models and Tools for Managing Development Processes



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Author

Bernhard Westfechtel
Department of Computer Science III, Aachen University of Technology
Ahornstr. 55, 52074 Aachen, Germany
E-mail: bernhard@i3.informatik.rwth-aachen.de

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Westfechtel, Bernhard:

Models and tools for managing development processes / Bernhard Westfechtel. -
Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ;
Singapore ; Tokyo : Springer, 1999

(Lecture notes in computer science ; Vol. 1646)

ISBN 3-540-66756-3

CR Subject Classification (1998): D.2, K.6, H.5.3

ISSN 0302-9743

ISBN 3-540-66756-3 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 1999

Printed in Germany

Typesetting: Camera-ready by author

SPIN: 10703812

06/3142 – 5 4 3 2 1 0

Printed on acid-free paper

Preface

The development of products in disciplines such as mechanical, electrical, or software engineering is a challenging task. Costs have to be reduced, the time-to-market has to be shortened, and quality has to be improved. Skilled engineers and sophisticated tools for supporting technical work are necessary prerequisites, yet they are not sufficient for meeting these ambitious goals. In addition, the work of developers must be coordinated so that they cooperate smoothly. To this end, the steps of the development process have to be planned, an engineer executing a task must be provided with documents and tools, the results of development activities have to be fed back to management which in turn has to adjust the plan accordingly, the documents produced in different working areas have to be kept consistent with each other, etc.

This book reports on models and tools for managing development processes. It provides both a survey of the current state of the art and presents our own contributions. The material covered in this book is based on research in different engineering disciplines (mechanical, software, and chemical engineering). It presents a unified view on the management of development processes in these disciplines.

The current state of the art is characterized by a large variety of tools for process management. Project management systems support classical management functions such as planning, organizing, and controlling by means of project plans (e.g., PERT charts). Engineering/product data management systems or software configuration management systems are concerned with the products of development processes such as designs, manufacturing plans, and NC programs in mechanical engineering; or requirements definitions, software architectures, and modules in software engineering. Workflow management systems manage the flow of work according to a defined procedure that coordinates activities such that defined objectives are achieved by set deadlines. Process-centered software engineering environments drive the software development process according to a process model that defines the steps to be executed and the constraints on their ordering.

Unfortunately, these tools still suffer from several limitations. Project management systems operate at a too coarse-grained level and do not take the products of development processes into account. Conversely, engineering/product data management systems and software configuration management systems focus on products, but neglect the management of activities. Workflow management

systems and process-centered software engineering environments are often too inflexible and do not adequately support the dynamics of development processes.

We present an integrated approach which addresses these shortcomings. We have developed models for managing development processes that consider products, activities, and resources in an integrated way. Moreover, the models are designed to cope with the dynamics of development processes. A development process may rarely be defined in advance; rather, it constantly evolves during execution. Some reasons are given below:

- The tasks to be performed depend on the product structure which is determined only during development. For example, the architecture of a software system defines the modules to be implemented and tested.
- Development rarely proceeds smoothly from one phase to the next. Rather, errors and inadequate solutions which are detected in later phases are fed back into earlier phases. The consequences of feedback may be hard to predict, and may range from small local changes to large global ones.
- “Walking on water and writing software from a specification are easy if both are frozen.” In reality, however, development must be prepared to cope with continuous changes to the requirements.
- In order to reduce development efforts, organizations strive for reusing previous results. Then, the development process depends on which results can be reused to what extent. This knowledge is often not available beforehand.
- If milestones have to be accomplished earlier than expected, it may be necessary to accelerate development on critical paths, assign more developers to the project, etc.
- Organizations are constantly striving for improving their processes resulting in optimized process definitions. It is desirable to propagate these optimizations into ongoing development processes.
- Current development methods such as concurrent and simultaneous engineering accelerate development by increasing parallelism. To be successful, they require the sophisticated coordination of engineers working on different parts of a product, or in different working areas such as design and manufacturing planning.

We have developed a management system which provides customized environments for its different kinds of users. The management environment supports managers in coordinating technical activities by presenting graphical, global views, and commands for planning, analyzing, controlling, and monitoring. Developers use the work environment which maintains agendas of tasks, manages a workspace of documents for each task, and offers a uniform interface for activating development tools in order to carry out technical activities. Finally, the modeling environment is used to adapt the management system to a specific application domain. So far, we have studied applications in mechanical and software engineering; our current work also addresses chemical engineering.

The data maintained by the management system and the operations performed on these data are fairly complex. This calls for a formal specification at

a high level of abstraction. We have selected attributed graphs as the underlying data model because they are ideally suited for representing management data such as version histories, configurations of interdependent documents, and task nets. A programmed graph rewriting system serves to specify operations on these graphs in terms of complex graph transformations. Management tools may be generated from this operational specification, avoiding the need for coding in a conventional, rather low-level programming language.

This book is composed of four parts. Part I introduces basic notions such as development, process, or management. Furthermore, it provides an overview of our approach to the management of development processes and compares it to related work.

Part II surveys the current state of the art. We draw a “grand picture” of models and tools for process management. To organize the discussion, we present taxonomies for classifying and comparing existing approaches. Furthermore, we apply these taxonomies to sample sets of process management systems in order to illustrate the spectrum of approaches developed in this field. Finally, we also attempt to assess the current state of the art.

Part III summarizes our work in SUKITS, an interdisciplinary project that was carried out by computer scientists and mechanical engineers at Aachen University of Technology. Its overall result was a management system which was applied to mechanical engineering within the project, but can be applied to other application domains as well. The management system supports integrated management of products, activities, and resources and takes various aspects of dynamics into account (in particular, product-dependent task nets, feedback, and simultaneous engineering). The management system was fully implemented, and it was successfully applied to non-trivial scenarios.

Part IV presents our ongoing work toward a universal and adaptable management model. This work was carried out in the final stages of SUKITS and subsequently in the IMPROVE project (a Collaborative Research Council dealing with development processes in chemical engineering).

This book is a revised version of my habilitation thesis. Many people have contributed to the work presented here. Prof. Manfred Nagl has been advising me for more than a decade. During this period, we have had many fruitful discussions; I have benefited much from his continuous inspiration. Prof. Carlo Ghezzi (Politecnico di Milano, Italy) and Prof. Theo Härder (University of Kaiserslautern, Germany) both agreed spontaneously to act as co-advisors in spite of their heavy workload.

My thesis was carefully reviewed for publication in the LNCS series. In particular, the review helped me considerably in improving the motivation for my work.

Prof. Andy Schürr has been a friend and colleague for a long time. My work on practical applications of graph rewriting is hardly conceivable without his contributions. In 1995, I spent a sabbatical at NTNU in Trondheim, Norway. This was the beginning of a fairly successful cooperation with Prof. Reidar Conradi

who provided me with many new insights into software configuration management. Several colleagues, students, and programmers have contributed to the work described in this book. In particular, I would like to thank Marita Breuer, Peter Heimann, Gregor Joeris, Dr. Carl-Arndt Krapp, Sven Krüppel, and Ansgar Schleicher. I would also like to thank all members of our group who have not been directly involved in my work. Each of them has assisted me in some respect, and they also created a good working atmosphere.

Finally, I would like to thank my wife Monika for her constant support and understanding. Moreover, I am indebted to my parents and my sister Anni. In particular, this book is dedicated to my father who has always supported and encouraged me.

Aachen, April 1999

Table of Contents

Part I. Introduction	
<hr/>	
1. Introduction	3
1.1 Development Processes	3
1.2 Management	5
1.2.1 Definition	5
1.2.2 Managerial and Technical Level	6
1.2.3 Products, Activities, and Resources	8
1.3 Tools for Managing Development Processes	10
1.3.1 Management System: Structure and Functionality	10
1.3.2 Scope of the Management System	12
1.4 Models for Managing Development Processes	12
1.4.1 Process Models	13
1.4.2 Models as Tool Specifications	13
1.5 Dynamics of Development Processes	14
1.5.1 Comparison to Other Business Domains	14
1.5.2 Dynamics at the Instance Level	15
1.5.3 Dynamics at the Definition Level	18
1.6 Approach and Contributions	19
1.6.1 Overall Approach	19
1.6.2 Context of Research	21
1.6.3 Models	23
1.6.4 Formal Specification	29
1.6.5 Tools	36
1.7 Related Work	41
1.7.1 Models	41
1.7.2 Formal Specification	43
1.7.3 Tools	45
1.8 Overview	48

Part II. Management of Development Processes: State of the Art

2. Process Management	53
2.1 Product Management	53
2.1.1 Documents and Configurations	54
2.1.2 Version Control	55
2.1.3 Workspace Management	57
2.2 Activity Management	57
2.2.1 Overview	57
2.2.2 Modeling of Activities	58
2.2.3 Functions of Activity Management	60
2.2.4 Tools for Activity Management	61
2.3 Resource Management	61
2.3.1 Human Resources	62
2.3.2 Computer Resources	63
2.4 Conclusion	64
3. Product Management	67
3.1 Functions of Product Management	67
3.2 Models for Product Management	69
3.2.1 Product Space	69
3.2.2 Version Space	71
3.2.3 Interplay of Product Space and Version Space	76
3.2.4 Construction of Versions	79
3.2.5 Workspaces	83
3.3 Tools for Product Management	85
3.3.1 Overview	85
3.3.2 System Descriptions	85
3.4 Conclusion	94
4. Activity Management	99
4.1 Universe of Discourse	100
4.1.1 Functions and Objectives of Process Management	100
4.1.2 Characterization of Development Processes	101
4.2 Models for Activity Management	104
4.2.1 Conceptual Framework	104
4.2.2 Process Meta Models	109
4.2.3 Product Development	112
4.3 Tools for Activity Management	117
4.3.1 Overview	117
4.3.2 System Descriptions	119
4.4 Conclusion	131

5. Resource Management	135
5.1 Management of Human Resources	135
5.1.1 Models for Human Resource Management	136
5.1.2 Tools for Human Resource Management	138
5.2 Management of Computer Resources	141
5.3 Conclusion	143
6. Tool Integration	145
6.1 Tool Integration: Classification and Overview	146
6.2 Tool Integration Technologies	149
6.3 Conclusion	153

Part III. A Management System for Mechanical Engineering

7. The SUKITS Project	157
7.1 Overview of the SUKITS Project	158
7.1.1 Motivation	158
7.1.2 Goals	160
7.1.3 Application Domain	162
7.1.4 Project Structure and Results	164
7.2 The SUKITS Approach to Process Management	165
7.2.1 Contributions	165
7.2.2 Management Models	166
7.2.3 Management Tools	167
7.3 Conclusion	168
8. Management Model: Informal Description	169
8.1 Product Management Model	169
8.1.1 Documents, Dependencies, and Configurations	170
8.1.2 Versions and Versioned Objects	170
8.1.3 Version and Configuration Graphs	172
8.1.4 Consistency Control and Data Integration	174
8.2 Activity Management Model	175
8.2.1 Product-Centered Activity Management	177
8.2.2 Process Dynamics	178
8.3 Resource Management Model	183
8.4 Integration of Formal and Informal Cooperation	185
8.5 Related Work	186
8.5.1 Product Management	186
8.5.2 Activity Management	189
8.5.3 Resource Management	191
8.6 Conclusion	192

9. Management Model: Formal Specification	193
9.1 PROGRES at a First Glance	193
9.2 Product Management Model	195
9.2.1 Graph Schema	195
9.2.2 Graph Transformations	204
9.3 Activity Management Model	214
9.3.1 Graph Schema	214
9.3.2 Graph Transformations	218
9.4 Resource Management Model	225
9.4.1 Graph Schema	226
9.4.2 Graph Transformations	227
9.5 Model Adaptation	229
9.5.1 Adaptation in SUKITS	229
9.5.2 PROGRES Specification of Model Adaptation	230
9.6 Discussion	234
9.6.1 Specification-in-the-Small	236
9.6.2 Specification-in-the-Large	237
9.7 Related Work	238
9.8 Conclusion	241
10. Management System	243
10.1 Tools: Functionality and User Interface	243
10.1.1 Management Environment	243
10.1.2 Modeling Environment	247
10.1.3 Work Environment	249
10.2 Realization	252
10.2.1 Management Environment	252
10.2.2 Modeling Environment	254
10.2.3 Work Environment	255
10.2.4 Tool Integration	256
10.2.5 Communication and Distribution	258
10.3 Conclusion	260
11. Applications, Experiences, and Evaluation	261
11.1 SUKITS Prototypes	261
11.2 Demonstration: Development of a Drill	264
11.2.1 Overview	264
11.2.2 Demo Steps	268
11.3 Evaluation	278
11.3.1 Management Model	278
11.3.2 Formal Specification	279
11.3.3 Management System	280
11.3.4 Applications	281
11.4 Conclusion	283

Part IV. Toward an Adaptable Environment for Modeling and Managing Development Processes

12. Overview	287
12.1 Conceptual Framework	288
12.1.1 Dimensions of Management	288
12.1.2 Model Architecture	290
12.1.3 Limitations of the SUKITS Approach	291
12.2 Modeling of Management Configurations	292
12.2.1 Architectural Issues	292
12.2.2 Models for Managing Products, Activities, and Resources	295
12.3 Management System	298
12.3.1 Overview	298
12.3.2 Modeling Environment	299
12.3.3 PROGRES Environment	301
12.3.4 Process Support Environment	302
12.4 Conclusion	302
13. Dynamic Task Nets	305
13.1 Informal Description	305
13.1.1 Structure of Task Nets	305
13.1.2 Levels of Modeling	308
13.1.3 Behavior of Task Nets	309
13.1.4 Examples	312
13.2 Formal Specification	315
13.2.1 Base Model	316
13.2.2 Standard Behavior	324
13.2.3 Structural Adaptation	326
13.2.4 Behavioral Adaptation	327
13.2.5 Discussion	328
13.3 Related Work	329
13.3.1 Net-based approaches	329
13.3.2 Rule-based approaches	330
13.3.3 State-based approaches	331
13.3.4 Procedural approaches	332
13.4 Conclusion	332
14. Unified Multi-Project Resource Management	333
14.1 Informal Description	333
14.1.1 Features of RESMOD	333
14.1.2 Resources	334
14.1.3 Resource Configurations	335
14.1.4 Plan and Actual Resources	335
14.1.5 Base and Project Resources	338
14.1.6 Task Assignments	340

14.2	Formal Specification	341
14.2.1	Layer 1: Resource Hierarchies	341
14.2.2	Layer 2: Actual and Required Resources	344
14.2.3	Layer 3: Base and Project Resources	350
14.2.4	Discussion	350
14.3	Related Work	353
14.4	Conclusion	354
15.	Object-Oriented Process Modeling	355
15.1	Motivation	355
15.2	Meta Process	356
15.3	Process Analysis	358
15.4	Process Specification	360
15.4.1	Structural Modeling	361
15.4.2	Model Structuring	363
15.4.3	Behavioral Modeling	365
15.5	Environment Generation	368
15.6	Lessons Learned	371
15.7	Related Work	373
15.8	Conclusion	374
16.	Current Status and Future Work	375
16.1	Management Model	375
16.2	Modeling Languages	376
16.3	Management System	377
16.4	Applications	377
16.5	Conclusion	378
A.	Glossary	379
	References	385
	Index	413