

Form-Oriented Analysis

Dirk Draheim · Gerald Weber

Form-Oriented Analysis

A New Methodology to Model
Form-Based Applications

With 83 Figures

 Springer

Dirk Draheim

Institute of Computer Science
Freie Universität Berlin
Takustr. 9
14195 Berlin, Germany
draheim@acm.org

Gerald Weber

Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland 1020, New Zealand
g.weber@cs.auckland.ac.nz

<http://www.formcharts.org/>

Library of Congress Control Number: 2004112522

ACM Computing Classification (1998): D.2.1, D.2.2

ISBN 3-540-20593-4 Springer 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, reuse of illustrations, recitation, broadcasting, reproduction on microfilm 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. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: KünkelLopka, Heidelberg
Production: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig
Typesetting: by the Authors
Printed on acid-free paper 45/3142/YL - 5 4 3 2 1 0

Preface

This book presents a special purpose modeling technique for the analysis and design of an important system class, namely form-based enterprise systems. Recent discussions on modeling languages emphasize that there is a strong demand for such domain-specific modeling languages. The class of form-based enterprise systems includes, for example, web shops as well as ERP and B2B solutions and can be said to be paradigmatic for enterprise computing. This book was motivated by the widespread interest in this type of business application from professionals as well as from scientists. The book adapts well-established basic modeling techniques in a novel way in order to achieve a modeling framework optimized for the indicated application domain.

Besides its practical parts the book details theoretical achievements, which lead to real improvements in the application domain of the book. It explains how to model a form-based enterprise system during the analysis and specification phase, and how these models translate into good design. Typical form-based applications have common properties that can be molded into specialized diagram types for such applications. Such a diagram type is the formchart, the central artifact that is described in the book. The formchart is a good example of customized diagrams according to the most recent proposed profiling techniques.

If form-based enterprise systems are modeled with typical general purpose modeling approaches without such customizations, there are a number of obstacles the modeler will face. For example, if one employs use case modeling together with interaction diagrams, the analyst will be confronted with three problems. First of all, the method has to be adapted to the form-based application, since no specific guidance for this special application type is part of the general method. Secondly, the model will become complex for even small-sized problems, since every diagram has to repeat common properties of this very specific application class. Hence the model tends to become highly redundant, and the important distinguishing information is diluted. Furthermore, a third problematic aspect is that current analysis methods are traditionally rather oriented towards event-driven and complex GUI-based applications but not

towards form-based applications. Hence the customization demand for form-based applications is particularly high.

Conversely, there are certain benefits the reader of the book can reap by employing the new customized artifacts presented in this book. The reader can obtain faster results and more significant models, because the common properties of enterprise systems are already incorporated in the semantics of the modeling method. The new artifact types presented in this book incorporate the results of studying form-based systems in general, knowledge that a software engineer can hardly obtain in the limited setting of a running project. Our method provides a separation of concerns by splitting the general semantic structure of such applications from the specific information about business logic in the concrete single project. The foundation of the new techniques is fully elaborated in the book for the working developer confronted with everyday problems in professional IT projects. In the same way, the scientist interested in performing novel research on enterprise systems can use this formal reference.

The book is divided into four parts. The first part is a detailed discussion of the new modeling method for form-based systems from a practitioner's viewpoint and explains how the proposed techniques can actually be employed in a project. The second part is about tool support and exemplifies how the concepts introduced in the first part can be exploited by several different implementing technologies. The third part provides the semantic foundation of the different kinds of diagrams and tools introduced in the first and second parts. The fourth part serves as a summary and provides a discussion of related work.

After the introduction the book starts with an in-depth motivation for the new techniques. It is shown that the considered system class encompasses a wide range of important enterprise systems from mainframe/terminal systems through ubiquitous COTS software to modern web applications. The explanations in the book are deliberately based on a realistic running example in order to make a difference. Throughout the book the concepts are exemplified with an online bookshop. This example is not an arbitrary choice of the authors – the important TPC-W benchmark, for example, also uses a standardized online bookshop as a representative example of typical business functionality. The form-oriented information system model is introduced. Different kinds of diagrams for these models, i.e., screen diagrams, page diagrams, form storyboards, and formcharts are introduced for the user interface state part of these models. All of these, and the further model components, i.e., dialogue constraints and the layered data models, are introduced immediately with unambiguous semantics and are used in modeling the running example. Then, techniques for decomposition and refinement are discussed. A parsimonious data modeling language is elaborated. A message approach to the modeling of data interchange is outlined. The book is not primarily about software engineering processes; however, it provides a discussion on how the proposed artifacts can be exploited in an entire software engineering life cycle. The

interplay of some proposed best practices that are centered around descriptiveness, artifact orientation, feature orientation, and reuse are discussed. For each concept we show how it can be used to add sustainable value to the respective software engineering activities.

The second part discusses issues of architecture, design, and implementing technology. From this discussion concepts and concrete prototypical technologies for forward engineering, reverse engineering, and the implementation of web presentation layers are derived.

The third part of the book presents the semantic foundation of form-oriented analysis. First, several alternatives for tool support are discussed and given a conceptual basis using an integrated source code model. Then, precise semantics for the form-oriented diagram types are given. For this purpose, a new, lightweight, semantics framework approach is introduced as an alternative to current multi-level metamodeling techniques. Along the lines of the framework approach precise semantics of formcharts, layered data modeling, and the dialogue constraint language are given. This is followed by a discussion of the semantic of the proposed parsimonious data modeling approach. A formal type system for the interplay of server actions and pages of submit/response style systems is provided.

The fourth part provides a focused description of the widely accepted modeling approaches in use. The discussion shows the differences between these approaches and the new method, but it also shows how our method is integrated with standard modeling techniques. For each related method we discuss how it could be applied to enterprise systems and how form-oriented analysis provides a more convenient solution. Therefore this chapter provides a different view on the benefits of form-oriented analysis to the reader. Finally, a summary of the main contributions is provided.

The reader should have some experience with object-oriented programming languages. First-hand experience with visual modeling languages and the graphical tools for them is helpful. Basic knowledge of SQL is also desired for some advanced excursions, but this can be postponed until needed. Related approaches are comprehensively introduced, so that even a reader who is new to these other approaches can follow the arguments.

The book targets professionals, i.e., working software engineers and decision makers, researchers in computer science, and upper-level graduate students who are interested in enterprise systems. Care must be taken, because professionals, researchers, and students typically have different objectives, different dispositions, and different opinions with respect to software engineering topics. This is due to the fact that goals and driving forces are different in industry and academia. Consequently, readers may have different attitudes towards the several parts of the book; see the figure below for a guess. In the figure, supposed main interests are shaded gray, whereas minor interests are left blank.

Professionals actually working on enterprise software will gain a deepened understanding of form-based systems from the abstract system viewpoint pro-

Part I Modeling Form-Based Systems	Part II Tool Support	Part III Semantics	Part IV Conclusion
Presentation of Form-Oriented Analysis	Practical Justification	Precise Reference Manual	Summary
Professional			
Presentation of Form-Oriented Analysis	Preliminary Semantics	Entry Point for Further Investigations	Discussion of Related Work
Scientist			
Presentation of Form-Oriented Analysis	Learning Aid		Summary
Student			

vided by form-oriented analysis. Many developers already use ad hoc techniques tailored to form-based systems like naive page diagrams or click dummies. These ad hoc techniques arise naturally when developing form-based systems but lack an elaborated conceptual basis. The book allows these developers to strengthen these techniques in practice. Readers can employ the approach directly in projects, because every concept introduced comes with precise semantics and the mutual dependencies between the concepts are elaborated, too. The prototypical forward and reverse engineering tools are suitable for convincing the professional about the potential practical impact of the form-oriented approach. The third part of the book is less important for the professional. If a semantical clarification is needed, this part can serve as a precise reference manual. The professional can use the fourth part as a detailed summary.

Researchers might be especially interested in the third part of the book as an entry point for further investigations. Upper-level graduate students will benefit from the presentation of state-of-the-art knowledge about the development, architecture, and design of enterprise systems in the organizing framework of form-oriented analysis. The second part of the book will help students to grasp more easily the concepts of form-oriented analysis.

Since enterprise applications are a particularly important class of software, almost every IT professional, computer scientist, or computer science student may have some interest in gaining at least an overview of the fundamentals of enterprise computing. The book is written with the different objectives of professionals, researchers, and students in mind. In industry productivity eventually targets return on investment. Product quality and product quantity are limited by productivity. Productivity is limited by the availability of resources. Knowledge acquisition is needed to improve productivity. Academic activity spans two areas that have to be integrated: research and education. While academic research has a subtle target, i.e., the construction of knowledge, higher

education has the tangible responsibility to produce well-prepared professionals. Academic research is driven by the pressure to get contributions published in the scientific peer community. Higher education is driven by the demands of the yet uneducated. Altogether these differences result in the following: one and the same concept can be perceived totally differently by individuals in industry and in academia. We encourage all who try to keep an open mind and hope that this book provides valuable information or inspiration.

We are indebted to Martin Große-Rhode for his encouragement and advice. We want to thank our editor Ralf Gerstner for his support and guidance. We would also like to thank the reviewers who made many helpful comments.

Berlin, August 2004
Auckland, August 2004

Dirk Draheim
Gerald Weber

Contents

Part I Modeling Form-Based Systems

1	Introduction	3
1.1	Enterprise Systems	4
1.2	Modeling Enterprise Systems	5
1.3	High-Level Transactional Programming	6
1.4	A Parsimonious Relational Notation	6
1.5	A Descriptive Approach to the Software Development Process	7
2	The Form-Based System Paradigm	9
2.1	The Submit/Response Style Interface	10
2.2	A Message-Based Model of Data Interchange	20
3	Exploring the Online Bookshop	23
3.1	The Dialogue Model	24
3.2	The Persistent Data Model	30
3.3	An Exemplary Submit/Response Style System	32
4	Form Storyboarding	35
4.1	Page Diagrams	36
4.2	Form Storyboards	41
5	Formcharts and Dialogue Specification	49
5.1	Form-Oriented Information System Models	49
5.2	The Dialogue Model	55
5.3	The Layered Data Model	59
5.4	Dialogue Specification	66
5.5	The Bookstore Formchart and Data Model	74

6	Model Decomposition and Refinement	97
6.1	Model Union	97
6.2	Formchart Decomposition	98
6.3	Formchart Hierarchies	100
6.4	A Feature-Driven Approach	102
6.5	State Sets and State Specialization	104
6.6	Decomposition of Page Diagrams and Form Storyboards	107
6.7	Model Refinement	107
7	Data Modeling	109
7.1	The Parsimonious Data Modeling Language	111
7.2	The Data Access Language DAL	114
7.3	The Transaction Data Access Language TDAL	121
7.4	Constraints	125
7.5	Style Formats	141
8	Message-Based Modeling of Data Interchange	147
8.1	Connectivity of Enterprise Systems	147
8.2	The Message-Based System Viewpoint	150
8.3	Data Interchange Model	152
8.4	Data Interchange Specification	156
8.5	The Relation to Data Flow Diagrams	160
8.6	The Interplay of Formcharts and Data Interchange Diagrams	162
8.7	Topic Bundles	164
9	A Descriptive Approach	165
9.1	Descriptiveness, Prescriptiveness, and the Software Process ...	168
9.2	Metaphor A Posteriori	170
9.3	On Desktop Metaphors	172
9.4	On Real-World Modeling	173
9.5	Visual Modeling De-emphasized	175
9.6	Artifact Orientation	176
9.7	The High-Level Programming Viewpoint	177
9.8	Advanced Systems Modeling Approaches	180

Part II Tool Support

10	Forward Engineering and Reverse Engineering	187
10.1	Forward Engineering	188
10.2	Reverse Engineering	193
10.3	Source-Code-Opaque Reverse Engineering	196

11	Typed Server Pages	199
11.1	Type-Safe Interplay of Forms and Scripts	200
11.2	Functional Decomposition of Server Pages	214
11.3	Higher-Order Server Pages	221
11.4	A Comparison of Web Technologies	224

Part III Semantics

12	The Integrated Source Code Paradigm	229
12.1	Towards Structured Collaborative Work	229
12.2	Structured Artifacts	232
12.3	The Syntax Model Approach	233
12.4	A Closer Look at Languages	235
12.5	The Integrated Source Code Paradigm	237
12.6	A Flexible Generic Textual Format for Data	241
12.7	Generative Programming	244
13	State History Diagrams	249
13.1	State History Diagrams and Class Diagrams	249
13.2	Discussion of Formchart Semantics	258
13.3	Semantics of Dialogue Constraints	260
13.4	Path Expressions	267
14	Semantics of the Data Model	271
14.1	Semantics of the Temporal Model	271
14.2	Alternative Fundamental Models	274
14.3	Discussion	279
15	Semantics of Web Signatures	283
15.1	Formal Semantics of Web Signature Recovery	283
15.2	Coding Guidelines for Typed Server Pages	287
15.3	Formal Definition of the NSP Type System	299

Part IV Conclusion

16	A Comparison of Modeling Methods	325
16.1	User Interface Modeling	325
16.2	Web Site Modeling	326
16.3	Data Modeling	328
16.4	Model-Oriented Specification Languages	329
16.5	Structured Analysis	331
16.6	Object-Oriented Analysis and Design	335
16.7	Model-Driven Architecture	338

17 Summary343

 17.1 Contributions to Modeling Form-Based Systems343

 17.2 Contributions to Modeling in General347

References351

Index367

List of Figures

2.1	Example pages of the online bookshop	10
2.2	Ultra-thin client based submit/response style systems	12
2.3	SAP R/3 system architecture	12
2.4	Example formchart for a system login capability	17
3.1	Welcome screen of the online bookshop	24
3.2	Login screen of the online bookshop	25
3.3	Registration screen of the online bookshop	26
3.4	Category screen of the online bookshop	27
3.5	Book page of the online bookshop	27
3.6	Shopping cart of the online bookshop	28
3.7	Order information page of the online bookshop	30
3.8	Search result page of the online bookshop	31
3.9	Data model of the online bookshop	32
4.1	Page diagram	37
4.2	Screen diagram	39
4.3	List of options for a single conceptual option	40
4.4	Form storyboard	42
4.5	Page images	44
4.6	Form storyboard annotated with interaction information	45
4.7	Message storyboard	47
5.1	The information system model of form-oriented analysis	50
5.2	A first formchart example	56
5.3	Two alternative formcharts for the same model	57
5.4	Formchart naming conventions	57
5.5	Opaque references in the information model	61
5.6	Formchart notational elements	67
5.7	Three-valued logic in OCL	68
5.8	Meaning of path expressions in the formchart	72

5.9	Complete formchart for the bookstore example	75
5.10	Login, registration and logout feature of the online bookshop ..	76
5.11	Conditional server/page transitions	79
5.12	Refinement of a server input constraint	80
5.13	Browsing feature of the online bookshop	85
5.14	Shopping cart feature of the online bookshop	88
5.15	Buying feature of the online bookshop	90
5.16	Reuse of the login and registration subdialogues	91
5.17	Managing user data in the online bookshop	92
5.18	Search feature of the online bookshop	93
5.19	Graphical representation of a client output constraint	96
6.1	Named partitioning of a formchart	99
6.2	Hierarchical formchart decomposition	100
6.3	Structured analysis: leveled data flow diagram	101
6.4	A flat formchart	101
6.5	Additional features of the bookstore	103
6.6	Modeling with state set notation	104
6.7	Using state set notation for the bookstore features	105
6.8	Modeling enabling conditions based on state specialization	106
7.1	The PD model of the bookshop	112
7.2	A submodel of the bookshop	116
7.3	A relation of arity 3	128
7.4	A partial order in the data model	129
7.5	A composition constraint	132
8.1	Example datatype interchange diagram	153
8.2	Model subsystem in a DTIM	154
8.3	Edges between actions within the same model	155
8.4	Example DTIMs of the login dialogue	160
8.5	A DTIM and an equivalent DFD	161
10.1	The Angie language related tool suite	187
10.2	Grammar of the language Angie	190
10.3	Bookstore login capability	192
10.4	Revangie: example screen classifications	198
11.1	CPDS and CPTS	201
11.2	Model 2 architecture	216
11.3	Model 2 architecture versus NSP functional decomposition	218
11.4	Example interaction diagram	221
11.5	Higher-order server pages design example	224
12.1	PD core syntax model	234
12.2	Multiplicity syntax model	234

12.3 A cutout message storyboard	243
13.1 Frameworks for state history diagrams and formcharts	252
13.2 A formchart is derived from the semantic framework	255
13.3 The object net over a formchart is a path	256
13.4 A login subdialogue as UML state machine	259
13.5 Semantics of path expressions in DCL	266
13.6 Example of path expressions	268
13.7 UML tree definition	269
15.1 Example form message type	293
15.2 Example cyclic form message type	296
15.3 Example complex form message type	297
15.4 List definitions	308
16.1 The Seeheim model of user interfaces	326
16.2 Data flow diagram	333
16.3 Use case diagram	337

Modeling Form-Based Systems