

Lecture Notes in Computer Science

2967

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

Springer

Berlin

Heidelberg

New York

Hong Kong

London

Milan

Paris

Tokyo

Sergey Melnik

Generic Model Management

Concepts and Algorithms



Springer

Series Editors

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

Author

Sergey Melnik
Microsoft Corporation
One Microsoft Way, Redmond, WA 98052-6399, USA
E-mail: melnik@microsoft.com

Library of Congress Control Number: 2004104636

CR Subject Classification (1998): H.3, H.2, D.2, D.3, F.3, I.2

ISSN 0302-9743

ISBN 3-540-21980-3 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 to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11007593 06/3142 5 4 3 2 1 0

Preface

Many challenging problems facing information systems engineering involve the manipulation of complex metadata artifacts, or *models*, such as database schemas, interface specifications, or object diagrams, and *mappings* between models. The applications that solve metadata manipulation problems are complex and hard to build. The goal of generic model management is to reduce the amount of programming needed to develop such applications by providing a database infrastructure in which a set of high-level algebraic operators, such as Match, Merge, and Compose, are applied to models and mappings as a whole rather than to their individual building blocks.

This dissertation presents an initial study of the concepts and algorithms for generic model management. We describe the first prototype of a generic model management system, introduce the algebraic operators that are used to manipulate models and mappings, clarify the semantics of the operators, and develop novel algorithms for implementing them. In particular, we present an innovative algorithm based on fixpoint computation that is used for implementing the generic operator Match, which finds correspondences between two models. Using the prototype and the operators presented in the dissertation, we develop solutions for several practically relevant problems, such as change propagation and reintegration.

April 2004

Sergey Melnik

Acknowledgements

I would like to express my deep gratitude to everyone who helped me shape the ideas explored in this dissertation, either by giving technical advice or encouraging and supporting my work in many other ways.

I enjoyed a rare privilege of collaborating closely with several distinguished database researchers. This dissertation would not have come into existence without their hands-on advice and motivation.

Prof. Erhard Rahm supervised and guided my work from the very first day. He gave me the opportunity to conduct this doctoral research and helped me make the right strategic decisions at many forks along the way. He kept me on track while allowing me to broaden my research horizon in tangential areas. His insightful comments, which densely filled the margins of each draft that I gave to him, gave rise to many creative ideas.

Prof. Hector Garcia-Molina invited me to Stanford University and taught me the art of turning hard research challenges into fun and expressing my thoughts clearly using examples. From him I learned that solid research requires patience: for example, he suggested that a draft of our joint paper [Melnik, Garcia-Molina, Rahm 2002] needed more polishing and so we missed a conference deadline. Later that paper, which underpins Part III of the dissertation, received the Best Student Paper Award at the Intl. Conf. on Data Engineering.

Prof. Emeritus Gio Wiederhold showed me what it takes to step back and see a big picture, and yet keep the details in focus. He gave me the opportunity to collaborate in the DARPA DAML project at Stanford and to get a foretaste of metadata management problems in the context of interoperation on the Semantic Web.

Dr. Philip A. Bernstein has been the driving force behind the emerging research area of generic model management, the subject of the dissertation. His vision papers and talks inspired much of the work done in this thesis. His insightful suggestions on our joint papers and his guidance in designing the first prototype for model management, which is presented in Part I of the dissertation, were invaluable.

Prof. Alon Halevy helped me keep my spirits high while I worked on Part II, a more theoretical part of the dissertation. His encouragement and advice made this work a real pleasure.

I am grateful to Profs. Serge Abiteboul, Paolo Atzeni, Stefano Ceri, Martin Kersten, Renée Miller, and Gerhard Weikum for helpful discussions.

I would like to thank my colleagues and friends in the Database Groups in Leipzig and Stanford, the members of the Graduate Programme on Knowledge Representation in Leipzig, and the colleagues in the RDF Core Working Group at the World-Wide Web Consortium for fruitful exchanges of ideas. The members of the Stanford Database Group helped me conduct the user study presented in Part III.

I am indebted to Drs. Stefan Decker, Andreas Paepcke, Bertram Ludäscher, Felix Naumann, and Arturo Crespo for their support and many informal discussions, which helped me put my academic research into perspective.

I owe very special thanks to my wife Teresa. Her love and energy constantly recharged my forces. She has been my perpetual source of creativity and inspiration, in so many respects.

This dissertation is dedicated to my parents, Tanja and Juri, who are truly the origin of all great things that ever happened to me.

The contributions of the above people made the work on this dissertation a rewarding and memorable experience. I thank you all.

April 2004

Sergey Melnik

Table of Contents

Part I. A Programming Platform for Model Management

1. Introduction	3
1.1 Metadata Management	3
1.2 The Problem	5
1.3 A Vision for Management of Complex Models	6
1.4 Outline and Contributions of the Dissertation	9
2. Conceptual Structures and Operators	13
2.1 Motivating Scenario	13
2.2 Conceptual Structures	18
2.2.1 Models	19
2.2.2 Morphisms	20
2.2.3 Selectors	22
2.3 Operators	22
2.3.1 Primitive Operators	23
2.3.2 Derived Operators	25
2.3.3 Extract and Delete	26
2.3.4 Match	27
2.3.5 Merge	28
3. Implementation and Applications	29
3.1 Conceptual Structures	29
3.2 Operators	30
3.2.1 Extract and Delete	30
3.2.2 Dependencies	32
3.2.3 ExtractMin	33
3.2.4 DeleteHard and DeleteSoft	35
3.2.5 Diff	36
3.2.6 Match	37
3.2.7 Merge	38
3.3 Prototype “Rondo”	42
3.4 View-Reuse Scenario	45
3.5 Reintegration Scenario	47
3.6 Conclusions	50

Part II. A Semantics for Model Management Operators

4. State-Based Semantics	55
4.1 Basic Concepts	56
4.1.1 Models	56
4.1.2 Mappings	58
4.1.3 Formal Notation	60
4.1.4 Semantics of Scripts	61
4.1.5 Preliminaries	62
4.2 Operators	64
4.2.1 Compose Operator	65
4.2.2 Invert Operator	67
4.2.3 Extract Operator	68
4.2.4 Merge Operator	73
4.2.5 Diff Operator	77
4.2.6 Confluence Operator	84
4.2.7 Match Operator	85
4.3 Materialization	86
5. Change Propagation Scenario	91
5.1 Propagating Additions	92
5.2 Propagating Deletions	93
5.3 A General Solution	95
5.4 Schema Evolution Scenario	96
5.5 Variants of Change Propagation	98
6. State-Based Semantics in Rondo	101
6.1 Semantics of Morphisms	101
6.2 Semantics of Selectors	105
6.3 Structural vs. State-Based Operators	106
6.4 Revisiting Change Propagation	109
6.5 Conclusions	112

Part III. Schema Matching

7. Similarity Flooding Algorithm	117
7.1 Overview of the Approach	119
7.2 Similarity Flooding Algorithm	122
7.2.1 Similarity Propagation Graph	122
7.2.2 Fixpoint Computation	123
7.3 Generalized Version of the Algorithm	124
7.4 Convergence and Complexity of the Algorithm	126
7.5 Features of the Algorithm by Example	127

7.5.1	Semistructured Data	128
7.5.2	XML Schemas	129
7.5.3	Matching XML Schemas Using Instance Data	131
7.5.4	Finding Related Data	134
8.	Filters	137
8.1	Constraints	138
8.2	Selection Metrics	139
8.3	FilterBest Algorithm	142
8.4	Expressing FilterBest in SQL	144
9.	Evaluation and Tuning	147
9.1	Matching Accuracy	148
9.2	Intended Match Result	150
9.3	User Study	151
9.4	Evaluation of Algorithm and Filters	153
9.5	Propagation Coefficients	155
9.6	Conclusions and Open Issues	156

Part IV. Model Management in Perspective

10.	Related Work	163
10.1	Data Integration and Merge	164
10.1.1	Schema Integration	165
10.1.2	Answering Queries Using Views	170
10.2	Schema Matching and Match	173
10.3	Mapping Composition and Compose	178
10.4	View Selection and Extract	181
10.5	View Complement and Diff	182
10.6	Approaches to Specifying Semantics	184
10.6.1	Semantics of Models and Mappings	184
10.6.2	Information Capacity	186
10.6.3	Category Theory	187
10.7	Metadata Repositories	189
10.8	Metadata-Intensive Applications	190
10.8.1	Declarative Mediation	190
10.8.2	Change Propagation	193
10.9	Other Related Work	195
11.	Conclusions and Outlook	199
11.1	Summary of Contributions	199
11.2	Concluding Discussion	200
11.3	Open Technical Challenges	205
11.3.1	Decidability and Complexity	205

- 11.3.2 Equivalence and Entailment of Scripts 205
- 11.3.3 Completeness and Redundancy..... 206
- 11.3.4 *N*-ary Mappings 209
- 11.3.5 Formalization of Model-Management Problems 210

- A. User Study 213**
 - A.1 BizTalk Schemas (XML) 214
 - A.2 Property Listing Schemas (XML)..... 215
 - A.3 Library Schemas (XML)..... 215
 - A.4 Product Schemas with Data Instances (XML)..... 215
 - A.5 University Schemas with Data Instances (XML) 216
 - A.6 Catalogs with Data Instances (XML) 217
 - A.7 Personnel Schemas (Relational) 219
 - A.8 University Schemas (Relational) 219
 - A.9 Personnel/University Schemas (Relational) 220

- B. Proofs of Simplification Theorems 221**
 - B.1 Extract Operator 221
 - B.2 Merge Operator 223
 - B.3 Diff Operator 225

- References 229**

List of Figures

1.1	A high-level architecture of model management	8
2.1	Scenario illustrating propagation of changes from a relational schema to an XML schema	14
2.2	Schematic representation of a solution for change propagation scenario of Fig. 2.1	15
2.3	Converted schema c and support element ORDERS in c'	16
2.4	Sample model shown as graph and 4-tuples	19
2.5	A morphism between a relational and an XML schema	21
2.6	Graph representation of XML schema in Fig. 2.5	21
2.7	Example of a selector	22
2.8	Examples of copying the model of Fig. 2.4 using selector $\{a1, a2, a3, a4\}$	26
3.1	Examples of extraction and deletion from a relational schema m ..	31
3.2	Example of existential dependencies in a relational schema	33
3.3	Example of existential dependencies in an XML schema	33
3.4	Merging two sample schemas	39
3.5	Architecture of the prototype	42
3.6	Code size breakdown in prototype (in lines of code)	45
3.7	Morphism between sources S_1 and S_2	45
3.8	Merging two SQL views	46
3.9	Reintegration scenario (3-way merge)	48
3.10	Schematic representation of the reintegration scenario	50
4.1	Some instances of relational schema R(Name: char(3), Sex: bool)	57
4.2	Portion of a mapping	58
4.3	Schematic representation for Example 4.2.6 (Extract)	68
4.4	Illustration of Extract operator	70
4.5	Schematic representation for Example 4.2.12 (Merge)	73
4.6	Illustration of Merge operator	74
4.7	Schematic representation for Example 4.2.17 (Diff)	77
4.8	Illustration of Diff operator	78
4.9	Example of Diff result by Theorem 4.2.5	80

4.10	The output mapping in Diff is not determined up to isomorphism	80
4.11	Illustration of Theorem 4.2.11 (Mirror Merge)	86
4.12	Materialization of models and mappings	88
5.1	Propagating additions	92
5.2	Propagating deletions	94
5.3	Propagating deletions over bijection	94
5.4	Change propagation: a general solution	96
5.5	Schema evolution: a special case of change propagation	97
5.6	Addition only, convert first then Diff	98
5.7	Addition only, Diff first, then convert	99
6.1	Three alternative semantics for a morphism	102
6.2	Relationship between cites and zip codes is not preserved on composition	104
6.3	Structural composition vs. state-based composition (the latter with and without NULLs; predicate $\leftrightarrow$ denotes if-and-only-if)	107
6.4	Structural extraction yields materialization of the state-based operator	108
6.5	Schematic representation for structural change propagation script	111
7.1	Matching two relational schemas: Personnel and Employee-Department	119
7.2	A portion of graph representation G_1 for relational schema S_1	120
7.3	Example illustrating the Similarity Flooding algorithm	122
7.4	Matching of semistructured data	128
7.5	Matching of two XML schemas: AccountOwner (S_1) vs. Customer (S_2)	130
7.6	Two different representations of XML data: OEM/Lore-like vs. XML/DOM-like	131
7.7	Matching of two XML schemas using instance data in DOM graph representation	133
7.8	Excerpt of relationships in the Stanford DB Group	135
8.1	Cumulative similarity vs. “stable marriage”	137
8.2	Relative similarities for the example in Fig. 8.1	138
8.3	Example illustrating execution of FilterBest in SQL	144
9.1	Matching accuracy as a function of t_{rel} -threshold for intended match results Sparse, Expected, and Verbose from Table 9.1	151
9.2	Average matching accuracy for 7 users and 9 matching problems	152
9.3	Matching accuracy for different filters and four versions of the algorithm	153

9.4	Impact of randomizing initial similarities on matching accuracy ..	155
9.5	Impact of different ways of computing propagation coefficients on overall matching accuracy in the user study	156
10.1	Use of composition in (Shanmugasundaram et al. 2001a)	180
11.1	Schematic representation for Conjecture 11.3.1 (Associative Merge).....	206
11.2	Illustration of Intersect operator	207

List of Tables

2.1	Summary of key operators in Rondo	23
2.2	Definitions of primitive operators	24
3.1	Comparison of variants of extraction and deletion	36
4.1	Summary of key model-management operators	64
7.1	A portion of <i>initialMap</i> obtained by string matching (10 of total 26 entries are shown)	120
7.2	The mapping after applying <i>SelectThreshold</i> on result of <i>SFJoin</i> .	121
7.3	Variations of the fixpoint formula	124
7.4	The mapping after applying <i>SFJoin</i> $\circ$ <i>SelectLeft</i> to semistructured data in Fig. 7.4	129
7.5	Parameters of the fixpoint computation for S_1 and S_2	131
7.6	Match results for XML schemas in Fig. 7.5 using two different graph representations	132
7.7	Match results for XML element tags in Fig. 7.7 using similarity threshold 0.05	134
7.8	Relatedness of faculty members in the DB group based on data in Fig. 7.8	135
9.1	Three plausible intended match results for matching problem in Fig. 7.1	149
9.2	Sizes of graphs in the user study	152
9.3	Illustration of convergence properties of variations of fixpoint formula for tasks $T_1, \dots, T_9$ in the user study. Shows iterations needed until length of residual vector got below 0.05.....	154
9.4	Different approaches to computing the propagation coefficients $\pi_{\{l,r\}}(\langle x, p, A \rangle, \langle y, q, B \rangle)$	157
10.1	Data integration scenarios	165