Ontologies for Software Engineering and Software Technology

Coral Calero · Francisco Ruiz · Mario Piattini (Eds.)

Ontologies for Software Engineering and Software Technology

With 84 Figures and 46 Tables

Springer

Editors

Coral Calero
Francisco Ruiz
Mario Piattini

E.S. Informática.
Paseo de la Universidad 4
13071 Ciudad Real, Spain

Coral.Calero@uclm.es
Francisco.RuizG@uclm.es
Mario.Piattini@uclm.es

Library of Congress Control Number: 2006932286

ACM Computing Classification (1998): D.2, H.1, I.2

ISBN-10 3-540-34517-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-34517-6 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

springer.com

© Springer-Verlag Berlin Heidelberg 2006

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.

Typesetting: by the Editors
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Cover: KünkelLopka, Heidelberg

Printed on acid-free paper 45/3100YL - 5 4 3 2 1 0

Preface

Overview

Two important challenges facing current communities of researchers and practitioners in the field of software engineering and technology (SET) are knowledge integration and computer-based automatic support. The first challenge implies wasting a lot of time and effort and this is due to one of the difficulties in human relationships, namely the lack of explicit knowledge shared among members of a group/project, with other groups and with other stakeholders. The second challenge arises because many projects include the design/construction of advanced tools for supporting different software engineering activities. These tools should provide as much functionality as possible with the smallest cost of development.

Both challenges can be better and more easily approached by using ontologies. In this book, we will mainly deal with two of the multiple applications of ontologies in software engineering and technology that have been identified in the literature: (1) sharing knowledge of the problem domain and using a common terminology among all the interested people (not just researchers); and (2) filtering the knowledge when defining models and metamodels.

The utility of the first application is obvious. However, it is important and convenient to pay it opportune attention. Communication is one of the main activities (regarding duration and impact) in software projects. It is proven that participants in projects have a different knowledge of the problem domain and/or use different languages. The ambiguity of the natural language implies mistakes and nonproductive efforts. Ontologies can mitigate these problems and, farther, some authors have intended to use ontologies as back-bone of software tools and environments.

The second application is focused on the filtering of knowledge of a given domain. Models and metamodels are abstract representations of reality and, by definition, they only include a part of the reality they are aimed at modeling, obviating the unwanted characteristics. In this sense, ontologies can also help us decide what must be extracted from the real systems

to build models or what must be taken into account when defining metamodels.

So, this book should not be considered as a book written by ontology experts for ontology experts, but one written by people who use the ontologies mainly for the two applications mentioned above. For that reason, this book is oriented to researchers and practitioners in SET and includes the advanced trends in the use of ontologies within software projects and software engineering research. It also deals with two main challenges the SET discipline: (1) knowledge integration and (2) design of more powerful and generic tools.

Organization

The book is composed of eleven chapters structured into three parts: an introductory part; a part composed of ontologies that conceptualize a SET domain or subdomain; and a part where some proposals on the use of ontologies as software artifacts in some software processes and technologies are described.

The last introductory part comprises two chapters. The first one, written by Oscar Corcho, Mariano Fernández-López and Asunción Gómez-Pérez, will introduce the ontologies' concepts and the main aspects related to ontological engineering. The second chapter (by Francisco Ruiz and José R. Hilera) will deal with the state of the art of the use of ontologies in SET. Also, this chapter defines a taxonomy for classifying the uses of ontologies in SET, together with the result of the classification into this taxonomy of about 50 ontologies (including the proposals of this book).

The second part is made up of five chapters. Chapter 3 will present the engineering of the ontology for the Software Engineering Body of Knowledge, written by Alain Abran, Juan-José Cuadrado, Elena García-Barriocanal, Olavo Mendes, Salvador Sánchez-Alonso and Miguel-Angel Sicilia. An ontology for software development methodologies and endeavours will be presented by Cesar Gonzalez-Perez and Brian Henderson-Sellers in Chap. 4. Chapter 5 presents a software maintenance ontology developed by Nicolas Anquetil, Káthia M. de Oliveira and Márcio G.B. Dias, and an ontology for software measurement by Manuel F. Bertoa, Antonio Vallecillo and Félix García is the topic of Chap. 6. An ontological approach to the SQL:2003 developed by Coral Calero and Mario Piattini will be explained in Chap. 7, closing this second part.

The final part begins with the Object Management Group Ontology Definition Metamodel (Chap. 8), developed by Robert Colomb, Kerry Raymond, Lewis Hart, Patrick Emery, Chris Welty, Guo Tong Xie and Elisa Kendall. Chapter 9, written by Uwe Assmann, Steffen Zschaler and

Gerd Wagner, deals with ontologies, metamodels and the model-driven paradigm. Chapter 10 will presents the use of ontologies in software development environments in the work of Káthia Marçal de Oliveira, Karina Villela, Ana Regina Rocha and Guilherme Horta Travassos. Finally, the topic of the last chapter of the book (Chap. 11) is a semantic upgrade and publication of legacy data by Jesús Barrasa Rodríguez.

As a complement to this book, the Alarcos Group (the research group of the editors) have created a web site (http://alarcos.inf-cr.uclm.es/ontoset) to store and share, in an open way and by using standardized formats, examples of interesting ontologies in the SET discipline. In addition to the examples referred to in the book, other examples of ontologies elaborated by the international community will be included in this web site.

Audience

The audience for this book is software engineering researchers and practitioners (professors, PhD and postgraduate students, industrial R&D departments, etc.). The reader is assumed to have previous knowledge of software engineering.

Acknowledgements

We would like to express our gratitude to all those individuals and parties who helped us produce this volume. In the first place, we would like to thank all the contributing authors and reviewers who helped improve the final version. Special thanks to Springer-Verlag and Ralf Gerstner, for believing in our project and for giving us the opportunity to publish this book. We would also like to thank José Carlos Villar Herrera and Tomás Martínez Ruíz of UCLM for their support during the developement of this book.

Finally, we would like to acknowledge the public organizations that financed this work, under the research projects CALIPO (TIC2003-07804-C05-03) and ENIGMAS (PBI-05-058).

Coral Calero
Francisco Ruiz
Mario Piattini

July 2006

Contents

1. Ontological Engineering: Principles, Methods, Tools and Languages 1
1.1 Introduction 1
1.2 What Is an Ontology? Viewpoints from a Philosopher and from an Ontology Engineer 3
1.3 What Are the Main Components of an Ontology? 5
1.4 Ontological Engineering 6
1.5 Principles for the Design of Ontologies 8
1.6 Ontology Development Process and Life Cycle 9
1.7 Methods, Methodologies, Tools and Languages 16
1.7.1 Methods, Methodologies and Tools Used for the Whole Ontology Development Life Cycle 16
1.7.2 Ontology Learning 22
1.7.3 Ontology Alignment and Merging 25
1.7.4 Ontology Evolution and Versioning 31
1.7.5 Ontology Evaluation 32
1.7.6 Ontology Implementation 34
1.8 Conclusions 38
1.9 Acknowledgements 39
References 39

2. Using Ontologies in Software Engineering and Technology 49
2.1 Introduction 49
2.2 Kinds of Ontologies 50
2.2.1 Heavyweight Versus Lightweight Ontologies 56
2.3 A Review of the Uses in SET 57
2.3.1 Ontology Versus Conceptual Model 63
2.3.2 Ontology Versus Metamodel 64
2.3.3 Ontologies in Software Engineering Environments 65
2.3.4 Representing Ontologies Using Software Engineering Techniques 67
2.3.5 Experiences and Lessons Learned in Software Engineering Research 69

2.4 A Proposal of Taxonomy 73
2.4.1 Ontologies of Domain 74
2.4.2 Ontologies as Software Artifacts 76
2.5 Review and Classification of Proposals in the Literature 79
2.5.1 Proposals of Ontologies of Domain 79
2.5.2 Proposals of Ontologies as Software Artifacts 86
References 95

3. Engineering the Ontology for the SWEBOK: Issues and Techniques 103
3.1 Introduction 103
3.2 History and Principles of the SWEBOK Project 105
3.2.1 Hierarchical Organization 107
3.2.2 Reference Material and Matrix 108
3.2.3 Depth of Treatment 108
3.3 The Ontology of the SWEBOK from a Conceptual and Consensus-Reaching Perspective 109
3.4 The Ontology of the SWEBOK as a Formal Artifact 112
3.5 Fundamental Elements of the Ontology of the SWEBOK 114
3.5.1 Activities, Artifacts and Agents 114
3.5.2 Models, Specifications and Methods 116
3.5.3 Theoretical Standpoints and Guidelines 117
3.6 Conclusions 119
References 120

4. An Ontology for Software Development Methodologies and Endeavours 123
4.1 Introduction 123
4.2 Ontology Architecture 125
4.2.1 The Communities Involved 125
4.2.2 Usage and Ontology Domains 127
4.2.3 Product and Process 131
4.3 Endeavour-Related Concepts 133
4.3.1 High-Level View 134
4.3.2 The Process Side 135
4.3.3 The Product Side 137
4.3.4 The Producer Side 140
4.3.5 Endeavour-Related Concepts: Conclusion 141
4.4 Method-Related Concepts 142
4.4.1 Templates and Resources 142
4.4.2 Duality in the Method Domain 143
4.4.3 Applying the Methodology 148

4.5 Conclusion 148
References 149

5. Software Maintenance Ontology 153
5.1 Introduction 153
5.2 Software Maintenance 154
5.3 An Ontology for Software Maintenance 156
5.3.1 Overview of the Ontology 157
5.3.2 The System Sub-ontology 158
5.3.3 The Computer Science Skills Sub-ontology 160
5.3.4 The Maintenance Process Sub-ontology 162
5.3.5 The Organizational Structure Sub-ontology 165
5.3.6 The Application Domain Sub-ontology 166
5.4. Validating the Ontology 166
5.4.1 Quality Validation 167
5.4.2 Relevance Validation 168
5.5 Putting the Maintenance Ontology to Work 169
5.6 Conclusion 171
References 172

6. An Ontology for Software Measurement 175
6.1 Introduction 175
6.2 Previous Analysis 177
6.3 A Running Example 178
6.4 The Proposal of Software Measurement Ontology 179
6.4.1 The SMO 179
6.5 Conclusions 194
References 195

7. An Ontological Approach to SQL:2003 197
7.1 Introduction 197
7.2 SQL Evolution 198
7.3 The Ontology for SQL:2003 201
7.3.1 The Data Types Sub-ontology 202
7.3.2 The Schema Objects Sub-ontology 204
7.4 Example 209
7.5 Conclusions 212
References 214

8. The Object Management Group Ontology Definition Metamodel 217
8.1 Introduction 218

8.2 Why a MOF Ontology Metamodel? 219
8.2.1 Why a Metamodel? 219
8.2.2 Why MOF? 220
8.2.3 Why Not UML? 221
8.3 The Ontology Development Metamodel 222
8.3.1 RDF/OWL Metamodel 224
8.3.2 Topic Maps 228
8.3.3 Common Logic 231
8.3.4 General Structure of Metamodels 233
8.4 Profiles and Mappings 235
8.4.1 The Need for Translation 235
8.4.2 UML Profiles 236
8.4.3 Mappings 238
8.4.4 Mapping CL 240
8.4.5 Interaction of Profiles and Mappings 241
8.5 Extendibility 242
8.5.1 Metaclass Taxonomy 242
8.5.2 Semantic Domain Models 243
8.5.3 *n*-ary associations 244
8.6 Discussion 244
8.7 Acknowledgments 245
References 246

9. Ontologies, Meta-models, and the Model-Driven Paradigm 249
9.1 Introduction 249
9.2 Models and Ontologies 253
9.2.1 What's in a Model? 253
9.2.2 What's in an Ontology? 255
9.3 Similarity Relations and Meta-modelling 257
9.3.1 Meta-models 258
9.3.2 Metameta-models 260
9.3.3 The Meta-pyramid, the Modelling Architecture of MDE 261
9.4 MDE and Ontologies 262
9.4.1 Domain and Upper-Level Ontologies 263
9.4.2 Relationship of Ontologies and System Models on Different Meta-levels 264
9.4.3 Employing Domain Ontologies in the MDA 265
9.4.4 Conceptual Benefits of an Ontology-Aware Meta-pyramid 267
9.4.5 Tools Based on an Ontology-Aware Meta-pyramid 268
9.4.6 The mega-Model of Ontology-Aware MDE 269
9.5 Related Work 270
9.6 Conclusions 271

9.7 Acknowledgments 271
References 271

10. Use of Ontologies in Software Development Environments 275
10.1 Introduction 275
10.2 From SDE to DOSDE 277
10.3 Domain-Oriented Software Development Environment 279
10.3.1 Domain Ontology in DOSDE 279
10.3.2 Task Ontology in DOSDE 280
10.3.3 Mapping Domain and Task 287
10.3.4 Using Knowledge Throughout the Software Development 288
10.4 From DOSDE to EOSDE 292
10.5 Enterprise-Oriented Software Development Environments 294
10.5.1 Enterprise Ontology 296
10.6 Tools in DOSDE and EOSDE 300
10.6.1 Domain Theory Browser 301
10.6.2 Sapiens: A Yellow Page's Software Tool 302
10.6.3 RHPlan: A Software Tool for Human Resource Planning 304
10.7 Conclusion 305
References 306

11. Semantic Upgrade and Publication of Legacy Data 311
11.1 Introduction and Motivation 311
11.2 Global Approach to Database-to-Ontology Mapping 314
11.3 Mapping Situations between Databases and Ontologies 315
11.4. The R_2O Language 319
11.4.1 A Mapping Description Specified in R_2O 320
11.4.2 Description of Database Schemas 321
11.4.3 Definition of Concept Mappings 322
11.4.4 Describing Conditions and Conditional Expressions 324
11.4.5 Describing Transformations 325
11.4.6 Attribute and Relation Mappings 326
11.5 The ODEMapster Processor 330
11.6 Experimentation: The Fund Finder Application 330
11.6.1 Ontologies in the Funding Domain 332
11.6.2 The Presentation Part: Semantic Publishing and Navigation 334
11.7 Conclusions and Future Work 335
11.8 Acknowledgements 337
References 337