

Ontology-based Application Integration

Heiko Paulheim

Ontology-based Application Integration

Foreword by Johannes Fürnkranz

 Springer

Heiko Paulheim
Knowledge Engineering Group
Technische Universität Darmstadt
Germany
paulheim@ke.tu-darmstadt.de

ISBN 978-1-4614-1429-2 e-ISBN 978-1-4614-1430-8
DOI 10.1007/978-1-4614-1430-8
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011936126

© Springer Science+Business Media, LLC 2011

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*This book is an extended version
of the dissertation
“Ontology-based Application Integration
on the User Interface Level”
at Technische Universität Darmstadt, D17.*

Foreword

There is probably no invention in the history of mankind that had such a profound impact on our lives in such a short time as the World Wide Web. Twenty years ago, Tim Berners-Lee has developed the first versions of HTML which allowed to weave documents into the large hypertext document that we know today. It was soon realized that the potential of this technology is not limited to connecting texts, but may serve as a backbone for a world-wide knowledge base called the *Semantic Web*. Again, Tim Berners-Lee helped to pioneer the vision of data and knowledge being publicly available in a formalized, machine-processable form. Based on standards like RDF or OWL, knowledge and semantics may be freely exchanged between heterogeneous applications. The number of facts stored in public knowledge repositories, so-called *ontologies*, is increasing at a rapid scale. Linked open data are on the verge of permeating our everyday lives.

Now we are facing the next revolution. Not only documents or knowledge will be connected, but computer applications are no longer running on personal computers, but on centralized servers which can be accessed via Web interfaces from a large variety of processors in smartphones, TVs, cars, household appliances, and more. For the end user, this not only relieves them of the burden of the update and maintenance of their software, but allows them to access their applications in a uniform way, everywhere and at every time.

A grand challenge for web-based software design is to integrate different heterogeneous applications into a homogeneous new system that utilizes the familiar existing components but allows a transparent data exchange between these components. Such *Mash-Ups* can be realized at the code level, by reprogramming functions of the individual applications, or at the data or business logic level by formalizing the service description and access of the applications, e.g. in the form of Web Services. Both ways have the disadvantage that aspects of the application have to be reprogrammed in order to allow a standardized data exchange.

This book shows how ontologies and semantic web technologies can be employed to solve the practical problem of integrating applications on the user interface level. It shows how the relevant concepts of user interfaces, such as components and interactions, can be captured in a highly formalized ontology, and puts a strong emphasis

on practical aspects of the implementation of an integration framework based on that ontology, such as the scalability of semantic event processing approaches, or the support of seamless cross-technological interactions. Never losing the focus on the end user, it further explores the possibilities ontologies provide to enhance the usability of integrated applications.

The book describes this innovative approach in all aspects. It provides an excellent introduction into ontologies and their applications in user-interface and application design, so that the book can be read without extensive prior knowledge in these areas. All presented concepts and techniques are illustrated with a case study that demonstrates the design of an integrated application for the management of catastrophes, which has been developed in a research project with different partners from the industry and academia, led by SAP Research. The book is thus of interest to both, researchers in ontologies who are looking for an interesting application, and for practitioners who want to find out techniques for combining different applications in a non-intrusive knowledge-based way. I am confident that it will become a key publication in its area.

Johannes Fürnkranz
Darmstadt, July 2011

Acknowledgements

While working on the topic of ontology-based application integration on the user interface level throughout the last three years, I have had valuable support and input from many people, without whom this book would not have become what it is.

Johannes Fürnkranz has encouraged me to pursue the topic of ontology-based application integration, agreed to take over the supervision of my dissertation, and has been continuously supporting me with valuable advice with respect to my research and numerous other topics. Jürgen Ziegler has agreed on acting as my secondary supervisor and has given me very valuable feedback in various discussions.

The topic covered by this book has been evolving out of and largely pursued within the research project *SoKNOS* at SAP Research in Darmstadt. First and foremost, Florian Probst has largely helped me in shaping the topic and taught me how to be formally precise when crafting ontologies in many fruitful (and sometimes exhausting) discussions. Many people both in Darmstadt and in other institutions involved in the SoKNOS project have been creating the atmosphere in which this research work could grow, especially Sebastian Döweling, Karen Tso-Sutter, Anna Lewandowski, and Thomas Ziegert, and everybody else who contributed to the SoKNOS project, in particular Daniel Oberle, Grigori Babitski, and Jörg Hoffmann at SAP Research Karlsruhe, Simon Bergweiler at DFKI, Alexander Walkowski, Christoph Stasch, and Florian Daiber at University of Münster, Stephan Braune and Martin Thoma at B2M, Marcus Hoffmann and Thorsten May at Fraunhofer IGD, and Alexander Behring and Andreas Petter at TU Darmstadt.

After the SoKNOS project was finished, I have continued pursuing my research in the *AdiWa* project and was happy to once more find a stimulating environment with great researchers, especially Benedikt Schmidt, Birgit Zimmermann, Christian Kuhn, Eicke Godehart, and Todor Stoitsev. During my whole time at SAP Research, Knut Manske, Nicole Scholl, and Anna Wypior have been supporting me with all the administrative work, Bettina Laugwitz has taught me interesting details about statistical evaluation that I had long forgotten about, and Andreas Faatz has been giving me valuable advice on very different issues far more than once.

During the work on this book, it was my great pleasure to supervise a lot of very talented students. Rene Abraham helped compiling an initial survey of user interface

description languages. Atila Erdogan worked on the implementation of the Flex container for Java user interfaces, and Lars Meyer evaluated different implementation alternatives for improving the performance of the integration framework, as well as he worked on the implementation and evaluation of the Semantic Data Explorer. Roland Plendl contributed to the implementation of the rule-based object exchange mechanism, and Tobias Wieschnowsky built a prototype of a graphical user interface integration tool.

For the prototype implementation of the approach discussed in this book, I have used the system *OntoBroker*, and some people supported me in getting non-standard features implemented and told me about secret configuration options which are not mentioned in the official manuals. At OntoPrise, Saartje Brockmanns and Roman Korf have been patiently answering my questions and revealed various hidden functionalities, and Michael Erdmann has pointed me to using Skolem terms for event processing rules.

It was already two years before I started to dive into the topic of ontology-based application integration that I began working with ontologies. Michael Rebstock and Janina Fengel have sparked my interest in that topic and worked with me for two years. Since then, I have been discussing research ideas with countless colleagues at workshops, conferences, and other occasions, and always gained a lot from those discussions.

Last, but not least, my wife Carolin and my family have been continuously supporting me during the last years. Thank you for everything.

Heiko Paulheim
Darmstadt, July 2011

Contents

1	Introduction	1
1.1	Vision	1
1.2	Challenges	3
1.3	Approach	4
1.4	Contributions	4
1.5	Outline of the Book	5

Part I System Integration, Ontologies, and User Interfaces

2	Application Integration on the User Interface Level	9
2.1	Fundamentals	9
2.1.1	Levels of Application Integration	9
2.1.2	Definition of User Interface Integration	11
2.1.3	Benefits of Application Integration on the User Interface Layer	12
2.1.4	Requirements and Challenges of Application Integration on the User Interface Layer	13
2.1.5	Design Space of User Interface Integration	14
2.2	Portals	15
2.3	Mashup Platforms	17
2.4	Other Commercial and Established Solutions	19
2.5	Prototypes from Academia	21
2.5.1	Snap-Together	21
2.5.2	OpenInterface Workbench	22
2.5.3	CRUISe	22
2.5.4	Mixup and mashArt	23
2.5.5	The Widget Composition Approach by Kotsalis	23
2.6	Identified Research Gaps	24
2.7	Summary	25

- 3 Ontology-based System Integration 27**
 - 3.1 What is an Ontology? 27
 - 3.1.1 Definitions 28
 - 3.1.2 Languages 28
 - 3.1.3 Reasoning on Ontologies 33
 - 3.1.4 Types of Ontologies 35
 - 3.1.5 Ontologies vs. Software Models 39
 - 3.1.6 Methodologies for Developing Ontologies 40
 - 3.2 Top Level Ontologies 42
 - 3.2.1 DOLCE 43
 - 3.2.2 SUMO 45
 - 3.2.3 Cyc 46
 - 3.2.4 PROTON 46
 - 3.3 Infrastructures for Working with Ontologies 49
 - 3.3.1 Ontology Editors 49
 - 3.3.2 Ontology Visualization 49
 - 3.3.3 Programming Frameworks 52
 - 3.3.4 Storage Solutions 54
 - 3.3.5 Ontology Matching 55
 - 3.4 Ontologies in Application Integration 55
 - 3.4.1 Data Source Level 56
 - 3.4.2 Business Logic Level 58
 - 3.4.3 User Interface Level 59
 - 3.5 Summary 59

- 4 Ontologies in User Interface Development 61**
 - 4.1 Classification Schema 61
 - 4.2 Ontologies for Improving Visualization 63
 - 4.2.1 Information Clustering 64
 - 4.2.2 Text Generation 64
 - 4.2.3 Adaptation of User Interface Appearance 65
 - 4.3 Ontologies for Improving Interaction 67
 - 4.3.1 Ontology-based Browsing 67
 - 4.3.2 User Input Assistance 68
 - 4.3.3 Providing Help 70
 - 4.3.4 Facilitating Interaction in Integrated User Interfaces 70
 - 4.4 Ontologies for Improving User Interface Development 71
 - 4.4.1 Identifying and Tracking Requirements 71
 - 4.4.2 Generating User Interfaces 72
 - 4.4.3 Reusing User Interface Components 73
 - 4.5 Summary 74

Part II Integrating User Interfaces with Ontologies

- 5 A Framework for User Interface Integration** 79
 - 5.1 Goals and Assumptions of the Approach 79
 - 5.1.1 Goals 79
 - 5.1.2 Assumptions 82
 - 5.2 Design Decisions 83
 - 5.2.1 Web-based vs. Non-web-based Approaches 83
 - 5.2.2 Centralized vs. Decentralized Communication 84
 - 5.2.3 Unified vs. Modularized Ontologies 85
 - 5.2.4 Open World vs. Closed World Semantics 87
 - 5.2.5 Hypothesized vs. Skolemized Instances 88
 - 5.3 Ontologies and Rules Used for User Interface Integration 89
 - 5.3.1 Ontologies for Event Annotations 89
 - 5.3.2 Ontologies for Describing Applications 91
 - 5.3.3 Ontologies for Capturing Applications’ States 92
 - 5.3.4 Integration Rules 92
 - 5.4 Basic Framework 94
 - 5.4.1 Roles and Tasks in User Interface Integration 94
 - 5.4.2 Technical Architecture 96
 - 5.5 Case Study: User Interface Integration in SoKNOS 101
 - 5.5.1 The SoKNOS Project 101
 - 5.5.2 Applications Integrated in the SoKNOS Project 103
 - 5.5.3 Example Interaction 105
 - 5.5.4 Special Cases 107
 - 5.5.5 Further Usage of Semantic Technologies in SoKNOS 112
 - 5.6 Summary 117

- 6 An Ontology of User Interfaces and Interactions** 119
 - 6.1 Existing User Interface Description Languages 119
 - 6.1.1 HTML and XForms 120
 - 6.1.2 XIML 122
 - 6.1.3 XUL 122
 - 6.1.4 TeresaXML and MARIA XML 123
 - 6.1.5 LZX 123
 - 6.1.6 WAI ARIA 125
 - 6.1.7 UIML 125
 - 6.1.8 UsiXML 126
 - 6.2 Design and Implementation of the Ontology 126
 - 6.2.1 Typical Queries 127
 - 6.2.2 Reused Generic Ontologies 128
 - 6.2.3 The Top Level 129
 - 6.2.4 The Detail Level 138
 - 6.3 Integration Rules 143
 - 6.3.1 Case Study: Definition of Integration Rules 144

6.3.2	Global Integration Rules	146
6.4	Summary	150
7	Data Object Exchange	151
7.1	Annotation of Class Models	151
7.2	Problems with Static Annotation	153
7.2.1	Multi-Purpose and Artificial Classes and Properties	153
7.2.2	Simplified Chains of Relations	154
7.2.3	Non-atomic Data Types	156
7.3	Non-Intrusive, Dynamic Annotation of Class Models	157
7.3.1	Rules for Mapping Class Models to Ontologies	157
7.3.2	Rules for Mapping Ontologies to Class Models	161
7.3.3	Object Exchange with Template-based Filtering	164
7.3.4	Non-intrusive Implementation	166
7.4	Case Study: Annotating the SoKNOS Class Model with the SoKNOS Domain Ontology	168
7.4.1	Annotation in SoKNOS	169
7.4.2	Mismatches in SoKNOS	169
7.4.3	Example Object Exchange in SoKNOS	171
7.4.4	Performance Evaluation	173
7.5	Summary	175
8	Efficient Semantic Event Processing	177
8.1	Why Efficiency is Important	177
8.2	State of the Art in Semantic Event Processing	178
8.3	Implementation Variants and Evaluation	180
8.3.1	Local vs. Global Event Processing	180
8.3.2	Pushing vs. Pulling of Instance Data	182
8.3.3	Local Caching of Instance Data	185
8.3.4	Design of Connector Rules	186
8.4	Generalization: Reasoning on Data From Running Applications	188
8.4.1	Generalized Architecture	188
8.4.2	Performance Evaluations	190
8.5	Case Study: Performance in SoKNOS	193
8.6	Summary	193
9	Crossing Technological Borders	195
9.1	Architecture for Cross-Technological User Interface Integration	195
9.1.1	Extending Containers	195
9.1.2	Handling Drag and Drop	197
9.1.3	Extending Information Objects	198
9.1.4	Combining Heterogeneous Applications with A-Box Connectors and Caches	198
9.2	Case Study: Integrating Java and Flex User Interface components in SoKNOS	199
9.2.1	Scenario	199

- 9.2.2 Implementing the Flex Container 200
- 9.2.3 Putting the Pieces Together 205
- 9.3 Summary 205

Part III The Future of Ontology-based UI Integration

- 10 Improving Information Exploration** 209
 - 10.1 Interaction with the Semantic Data Explorer 209
 - 10.2 Architecture 211
 - 10.3 Evaluation 212
 - 10.3.1 Scenario 212
 - 10.3.2 Evaluation Setup 213
 - 10.3.3 Evaluation Results 215
 - 10.4 Summary 218
- 11 Towards End-user User Interface Integration** 219
 - 11.1 Current Approaches to UI Integration Revisited 219
 - 11.2 Enabling End Users for Ad Hoc Integration 222
 - 11.3 Towards a Visual, Ontology-based Tool for User Interface Integration .. 224
 - 11.4 Summary 226
- 12 Conclusion and Outlook** 227
 - 12.1 Achievements 227
 - 12.2 Summary of Evaluations 228
 - 12.3 Open Research Issues 230
 - 12.3.1 Improving the Development Process of User Interface Integration 230
 - 12.3.2 Enhancing the Capabilities of Integrated User Interfaces ... 231
 - 12.3.3 Possible Influence on Related Fields 232
 - 12.4 Concluding Remarks 233
- List of Abbreviations** 235
- List of Figures** 239
- List of Tables** 243
- References** 245
- Index** 267

