SEMANTIC MANAGEMENT OF MIDDLEWARE

SEMANTIC WEB AND BEYOND
Computing for Human Experience

Series Editors:

Ramesh Jain
University of California, Irvine
http://jain.faculty.gatech.edu/

Amit Sheth
University of Georgia
http://lsdis.cs.uga.edu/~amit

As computing becomes ubiquitous and pervasive, computing is increasingly becoming an extension of human, modifying or enhancing human experience. Today's car reacts to human perception of danger with a series of computers participating in how to handle the vehicle for human command and environmental conditions. Proliferating sensors help with observations, decision making as well as sensory modifications. The emergent semantic web will lead to machine understanding of data and help exploit heterogeneous, multi-source digital media. Emerging applications in situation monitoring and entertainment applications are resulting in development of experiential environments.

SEMANTIC WEB AND BEYOND
Computing for Human Experience
addresses the following goals:

- brings together forward looking research and technology that will shape our world more intimately than ever before as computing becomes an extension of human experience;
- covers all aspects of computing that is very closely tied to human perception, understanding and experience;
- brings together computing that deal with semantics, perception and experience;
- serves as the platform for exchange of both practical technologies and far reaching research.

Additional information about this series can be obtained from
http://www.springer.com

SEMANTIC MANAGEMENT OF MIDDLEWARE

by

Daniel Oberle
University of Karlsruhe, Germany

Springer

Daniel Oberle
University of Karlsruhe
Institute of Applied Informatics and
Formal Descriptions Methods
D-76128 Karlsruhe
Germany

Library of Congress Control Number: 2005908104

Semantic Management of Middleware
by Daniel Oberle

ISBN-10: 0-387-0-387-27630-0 e-ISBN-10: 0-387-0-387-27631-9
ISBN-13: 978-0-387-27630-4 e-ISBN-13: 978-0-387-27631-1

Printed on acid-free paper.

Dissertation, genehmigt von der Fakultät für Wirtschaftswissenschaften der Universität Fridericiana zu Karlsruhe, 2005. Referent: Prof. Dr. Rudi Studer, Korreferenten: Prof. Dr. Bruno Neibecker, Prof. Dr. Steffen Staab

Cover art by Daniel Oberle, showing the architecture of the ontology-based application server, from another perspective. The green object is a common symbol for an ontology.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

springeronline.com

Dedicated to ...

... the Supervisors

Steffen Staab
Andreas Eberhart
Pascal Hitzler
Rudi Studer

... the Reviewers

Ed Curry
Robert Delaney
Aldo Gangemi
Peter Haase
Stefan Tai
Denny Vrandecic
Clare Waibel

... the Colleagues @ Semantic Karlsruhe

Andreas Abecker, Sudhir Agarwal, Stephan Bloehdorn, Saartje Brockmans, Philipp Cimiano, Christian Drumm, Marc Ehrig, Stephan Grimm, Siegfried Handschuh, Jens Hartmann, Andreas Hotho, Steffen Lamparter, Jens Lemcke, Alexander Mädche, Boris Motik, Kioumars Namiri, Gisela Schillinger, Lars Schmidt-Thieme, Christoph Schmitz, Ljiljana & Nenad Stojanovic, Gerd Stumme, York Sure, Julien Tane, Bernhard Tausch, Christoph Tempich, Max Völkel, Johanna Völker, Raphael Volz, Susanne Winter, Valentin Zacharias

... the Colleagues @ World

Sean Bechhofer, Bettina Berendt, Jorge Gonzalez, Nicola Guarino, Frank van Harmelen, Ian Horrocks, Hans-Arno Jacobsen, Holger Knublauch, Deborah McGuinness, Sheila McIlraith, Peter Mika, Christof Momm, Doug Lea, Jeff Pan, Helena Sofia Pinto, Debbie Richards, Marta Sabou, Jorge Santos, Rick Schantz, Luc Schneider, Swaminathan Sivasubramanian, Heiner Stuckenschmidt, Phil Tetlow, Peter Spyns, Mike Uschold, Werner Vogels, Frank Wolff

... and last but not least to my family & friends!

Contents

List of Figures

List of Tables

Preface

We have termed this series "Semantic Web and beyond: computing for human experience." Ramesh Jain (co-editor of this series) and I believe that semantics is going to be far more pervasive than portrayed by the current vision of the Semantic Web. Its role and values will certainly not be limited to the traditional Web. Semantics will also be one of the important components of a continuum leading to perception and experience, albeit one that will mature earlier in computational context. We also believe that computation, supported by techniques and technologies that deal with perception, semantics, and experiences, will improve and benefit human experience. Such a computation will have a far broader impact than the traditional drivers of information technology, such as improving efficiency, lowering cost, or productivity gains. In this context, we expect that this series intends to offer additional books covering topics in perception, semantics, and experiential computing as they relate to improving human experience involving interactions with computing devices and environments. Our series intends to offer research monographs, books for professional audiences, as well as text books for advance graduate courses.

This premier book in our series by Daniel Oberle is a good example of what we hope to cover in this series. It discusses the role of semantics in middleware — arguable the most important segment of the enterprise software market. This work demonstrates that semantics and the semantic (and Semantic Web) technologies have pervasive applications and uses. It is also an excellent training companion for active practitioners seeking to incorporate advanced and leading edge ontology-based approaches and technologies. It is a necessary preparation manual for researchers in distributed computing who see semantics as an important enabler for the next generation.

Middleware systems are complex. They need to integrate and manage multiple heterogeneous software systems. Just as semantics has been recognized as a key enabler of heterogeneous information integration, can semantics be a key enabler in integrating heterogeneous software systems? Daniel believes

that is indeed the case, and he goes on to provide a detailed road map on how semantics and Semantic Web technologies can play a significant role in creating a middleware system and their use in managing heterogeneous software systems.

The first step in the road map is modelling which centers around using ontologies for specifying semantic models. This leads to the development of a semantic model for software components and Web services. We are introduced to the basics of middleware technology where technologies and design patterns from the past, such as message-oriented middleware or object monitors, help readers who are not familiar with the area to get the necessary background information. The discussion on ontologies achieves the same.

The subsequent part of this book offers a detailed discussion on the different ontology frameworks which can be used as a modelling basis. The requirements for the ontology are well laid out and each ontology framework is analyzed with respect to the requirements. DOLCE is chosen since it meets most of the requirements. The discussion on the semantic modelling of software systems is particularly interesting to read. The author addresses the different modelling issues at different phases of software component design. The modelling captures the intricate details and differences between fundamental concepts, such as data and software, and continues with component profiles, policies, and many more aspects. The modelling also captures the API of components and proposes a technique to discover dependent and conflicting libraries. Also presented is a model to capture workflows. A discussion on how such a modelling can meet the requirements and the advantages of using such an approach are presented in detail.

The next part provides a technical look at the different solutions. This includes discussing each aspect of the middleware system and the techniques to realize them as actual systems. The requirements of such a semantic middleware are presented and the system design is discussed with that central perspective. The system architecture along its different constituting components are discussed in detail. The application and reuse of the proposed ontology in the middleware system is also presented. The book ends with relating this approach to application management, Semantic Web Services and MDA.

Potentially, the most lasting engineering progress in this book, in my personal view, is that of taking semantics to the application server level. I foresee an emergence of Semantic Aware Networking, in which semantics not only facilitates network functions but significantly enhances its capability by pushing more functions. With the industry already taking initial steps in this direction, such as in CISCO's Application Oriented Networking products, the next step is quite likely the interplay between routers and application servers with semantics providing a bridge.

While there is plenty of work related to semantics of information and even Web services, this effort stands out in its attention to modelling the semantics of software components. In this context, it is a unique offering that goes beyond the mainstream Semantic Web research, while demonstrating a detailed and pervasive use of semantics in larger software systems context. This series will endeavor to offer more such books and for wider audiences.

Amit P. Sheth
Director, Large Scale Distributed Information Systems lab
Professor, Computer Science Department, University of Georgia
CTO and Cofounder, Semagix, Inc.
Athens, Georgia,
U.S.A.

Foreword

Which topic in computer science has been attracting researchers *and* developers from *artificial intelligence*, *business process modelling*, *conceptual modelling*, *databases*, *distributed systems*, *information systems*, *programming systems*, *security*, *software engineering*, *Web services* and *Web systems and engineering* (and probably many others whom I forgot to mention here)? It is the specification, development and management of component- and service-oriented architectures (SOAs).

The topic has become important to all of them. While the development of individual software systems is reasonably well understood and reasonably well-established practice (even when thousands of issues of such systems are and will have to be dealt with in more detail), the specification, development and maintenance of distributed software systems has in generalnot been understood to an extent that let their stakeholders gain the intended economic network benefits. On the upside, organizations that establish networking of their numerous distributed systems with the ones of other organizations may save costs, gain new customers or increase customer satisfaction. On the downside, if each minor change or minor disruption in one software system leads to a trickle down effect that results in expensive reprogramming of another system, all the potential positive network effects are overshadowed by the costs for joining and remaining in the network.

As the spectrum of interests indicates, the solution to this dilemma may require a multi-faceted approach. This book by Daniel Oberle significantly contributes towards this objective. Hence, the methods he proposes, revises and extends contribute to the plentiful, seminal research of several communities. I will name the ones that are most immediately affected, though I strongly believe that all of the above cited interest groups may benefit from building on his contribution:

Distributed Systems: Current middleware systems, such as application servers, are complex beasts that are very hard to tame because of the intricacies of distributed interactions. Hence, it has been a long-established practice to factorize configuration aspects of distributed interactions into corresponding declarative description files and — more recently — into XML files that follow the specification given by the various Web service standards.

Unfortunately, the semantics of these files is either given by the code of the concrete middleware system or — probably worse — by thousands of pages of specification documents consisting of raw textual explanations. We all know what went wrong with compilers in the 1960ies when programming language specifications were still at that stage.

Daniel here builds a rigorous approach towards giving the declarative descriptions of software components/Web services a well-defined meaning by defining ontological foundations *and* by showing how such foundations may be realized in practical, up-and-running systems.

Artificial Intelligence — Ontologies: Though all software developers use programming languages, only few specialists are actually able to formally define a programming language and develop a corresponding compiler: the formal foundation is not used to tutor the beginner, but to clarify the discussion and development by experts. The same is true for ontologies that underly a software management approach. They need to outlive many software development cycles, i.e., they need to have a formal foundation, yet one must also tutor the domain experts how to use them.

It is one of the successes of this work that it shows how to develop *and* use the ontological foundations of this work in a concrete software environment. This is done in a way that the usage of the resulting middleware infrastructure seems amenable to a sophisticated software developer even though the development of a complex foundational ontology may have to be left to some few specialists.

Web Services — Semantic Web Services: The analysis of the ontologies Daniel develops makes evident that *very few concepts* actually differ when "upgrading" from conventional middleware to Web services. It also makes clear that the use of declarative specifications, such as done in Web services, or formal declarative specifications, such as done for Semantic Web Services comes with economic modelling costs that need to be justified by savings in other places. This lets us presume that formal specifications with the objective of fully automatic Web service composition and orchestration remain a valid research topic — but one that will find its applications in niches rather than in wide-spread adoption by software developers.

Thus, the book covers an incredible depth and breadth of approaches. Its value lies in revising and extending existing methods thereby providing the cornerstones for specifying, developing and managing distributed applications in the coming decades — using semantics.

Prof. Dr. Steffen Staab
ISWeb — Information Systems and Semantic Web
Institute for Computer Science
University of Koblenz-Landau
Germany

Acknowledgments

This work was financed by WonderWeb — "Ontology Infrastructure for the Semantic Web," a European Union Information Society Technologies (IST) Future Emerging Technolgiges (FET) funded project under contract number IST-2001-33052 (2002-2004). I feel indebted to all the colleagues for the much valued cooperation and fruitful discussions we had throughout the project.

`http://wonderweb.semanticweb.org`

This work was financed by SmartWeb, a research project funded by the German Federal Ministry of Education and Research (BMBF). My appreciation to all the colleagues of another great project still running at the time of writing this book (2004-2007). It has been a joy working with you all!

`http://smartweb.semanticweb.org`

My appreciation to all the members of the Software Engineering Task Force (SETF). The task force is a part of the World Wide Web's Consortium (W3C) Semantic Web Best Practices and Deployment Working Group (SWBPD), where this work was also successfully exposed and revised [Tetlow et al., 2005].

`http://www.w3.org/2001/sw/BestPractices/SE/`

Finally, I would like to express my gratitude to Amit Sheth — the series editor — as well as the ladies at Springer: Susan Lagerstrom-Fife and Sharon Palleschi, who made the publication of this book possible.