

Foreword

The first time I paid attention to the term “ontology” was in the late 1980s when I was part of an engineering team that was responsible for defining what we would now call a domain-specific modeling language. In our case, the domain was telecommunications software and the purpose of our language was to give system architects the ability to describe the high-level structure of their software in the most direct and most expressive manner possible.

The team members were all experienced designers with deep knowledge of the domain so that we had no trouble putting together the initial list of key language concepts. We knew that we needed to include standard architectural modeling constructs such as components, ports, connectors, and the like. We also wanted our language to be object-oriented, so notions such as class, objects, and inheritance were added to the list. However, soon after this very promising start, all progress ground to a halt. Somehow, the definition of the seemingly trivial fine-grain details of these constructs kept eluding us despite long, passionate, and occasionally acrimonious discussions that can only be compared to medieval theological debates.

It was our good fortune that at that point we met Professor Doug Skuce of the University of Ottawa. He had a method and a tool that helped us develop an explicit ontology for our domain. From that exercise we learned that our difficulties stemmed from the fact that, although we shared a general intuition for the chosen constructs of our language, there were numerous subtle and *unstated* differences in our individual conceptualizations that were a barrier to mutual understanding. Furthermore, we discovered that certain commonly used terms had multiple meanings – all equally valid – but which we had not differentiated adequately, leading to much confusion. Only after we had defined our ontology, which included semi-formal definitions of all key terms and their relationships, were we able to finish our task successfully.

Ever since I’ve felt that defining a formal domain ontology is a useful and often necessary step in almost any software project. This is because software deals principally with ideas rather than physical artifacts. Whereas the nature of physical artifacts is generally self-evident, this is not

the case with conceptual entities, which are products of the mind. As we all know, different minds see the same thing differently.

The definition and application of ontologies for developing software systems is a central theme of this book. However, the book is about much more than that. It explains, in a clear and didactic manner, how a variety of recent buzzword developments in software theory and practice (intelligent agents, Model Driven Architecture, metamodeling, etc.) can be combined, and brings us to the threshold of the next step in the evolution of the World Wide Web: the *Semantic Web*. Like the Internet before it, the Semantic Web promises to introduce a significant and qualitatively new phenomenon into our lives. This is because it endows the network of disparate information that is currently accessible on the Internet with *meaning*. Because this meaning can be gleaned and processed *automatically* by software, the Semantic Web opens up the exciting and awe-inducing possibility of a unified global intelligence accessible to all.

In the first half of the book, the authors navigate deftly through the prolific and highly confusing *gemischt* of technologies, tools, and standards such as XML, RDF, OWL, MDA, or UML and explain how they relate to each other in the context of the idea of the Semantic Web. They introduce the notion of *modeling spaces*, which provides a conceptually simple yet comprehensive framework for understanding and addressing issues within the domain considered. Using that framework, the second half of the book describes a practical strategy for realizing key elements of the Semantic Web based on existing industry standards.

The book is equally suited to those who merely want to be informed of the relevant technological landscape, to practitioners dealing with concrete problems, and to researchers seeking pointers to potentially fruitful areas of research. The writing is technical yet clear and accessible, and is illustrated throughout with useful and easily digestible examples.

I would also highly recommend this book to sociologists studying the interplay between society and technology. It clearly demonstrates that the core technologies required for constructing the Semantic Web are available and moving forward inexorably. Society must be prepared to deal with something so ripe with potential. We must understand not only how the Semantic Web can be useful but also what dangers lurk within it.

Ottawa, Canada

Bran Selic

December 2005