Modeling & Simulation-Based Data Engineering
Modeling & Simulation-Based Data Engineering:

Introducing Pragmatics into Ontologies for Net-Centric Information Exchange

Bernard P. Zeigler
Phillip E. Hammonds
This book owes much to a multitude of colleagues whose ideas and hard work contributed to its development. Among those that come to mind are: Dasia Benson, Steve Bridges, Erika Carswell, Saehoon Cheon, Rich Clarke, Jerry Couretas, Robert Flasher, Dale Fulton, Anthony Galassi, Steve Gayer, Xiaolin Hu, Moon Hwang, Steve Kerr, Doowhan Kim, Ken Kingston, Rodney Leist, Stephen Madden, Eddie Mak, Saurabh Mittal, Robin Moore, Kimberly Nunn, James Nutaro, Greg Plum, Allan Reifer, Hessam Sarjoughian, Chad Schulenberg, Chungman Seo, and Ming Zhang. To them, and all others we may have failed to mention, we offer our heartfelt appreciation.

Bernard P. Zeigler
Phillip E. Hammonds
Arizona, 2007
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FOREWORD</strong></td>
<td>xi</td>
</tr>
<tr>
<td><strong>PART I  FOUNDATIONS</strong></td>
<td></td>
</tr>
<tr>
<td>1  ONTOLOGIES AND INFORMATION EXCHANGE: A PRAGMATIC FRAMEWORK</td>
<td>3</td>
</tr>
<tr>
<td>How Ontologies generate world state descriptions for use in application contexts called pragmatic frames</td>
<td></td>
</tr>
<tr>
<td>2  BACKGROUND: SYNTAX, SEMANTICS, AND ONTOLOGY FRAMEWORK</td>
<td>13</td>
</tr>
<tr>
<td>Introduction to XML and related ontology environments</td>
<td></td>
</tr>
<tr>
<td>3  FORMULATING PRAGMATIC FRAMES AND ONTOLOGIES: GEOSPATIAL SENSOR DATA</td>
<td>35</td>
</tr>
<tr>
<td>Illustrating pragmatic frame concepts in the geospatial sensing context</td>
<td></td>
</tr>
</tbody>
</table>
PART II  SYSTEM ENTITY STRUCTURE CONCEPTS AND OPERATIONS

4  INTRODUCTION TO THE SYSTEM ENTITY STRUCTURE  53
Basic concepts of system entity structure (SES) with examples of how to apply them

5  SYSTEM ENTITY STRUCTURE AXIOMS: INTERPRETATIONS AND APPLICATIONS  71
Illustrated presentation of the set theoretic formulation of SES and how the axioms are interpreted and how they impact application

6  SYSTEM ENTITY STRUCTURE: COMPUTATIONAL REPRESENTATIONS  90
Representing SES in XML, DOM, JAVA, and Natural Language and their transformations

7  MAPPINGS: TRANSFORMATIONS AND RESTRUCTURINGS  113
Restructurings of the SES that support development and harmonization

8  PRUNED ENTITY STRUCTURES AND XML SCHEMA INSTANCES  128
Pruning as a process for generating world state descriptions and their encodings in XML

9  CONSTRAINED PRUNING  146
Using rules and relations to constrain pruned entity structures (PES) to better describe the application domain

10  PRUNED ENTITY STRUCTURES: DATA EXTRACTION AND CHANGE-BASED INFORMATION EXCHANGE  176
Applying the SES and PES structures to implement efficient data extraction and information exchange
## PART III  MODELING AND SIMULATION AND DATA ENGINEERING

### 11  HIERARCHICAL SYSTEMS, MODELS, AND SIMULATIONS: THE SES ONTOLOGY  195
Connects the SES with the domain of simulation modeling using application examples

### 12  MANAGING SYSTEM ENTITY STRUCTURES: COMPOSING LARGE SYSTEMS  210
Merging SESs and PESs to compose large hierarchical structures from components

### 13  HARMONIZING DATA REPRESENTATIONS AND ONTOLOGIES WITHIN PRAGMATIC FRAMES  219
How to use Pragmatic Frames and SES concepts to provide a sound basis for information exchange

### 14  GEOSpatial SENSOR DATA: THE UNIVERSAL PHASE HISTORY DATA (UPHD) STANDARD  242
An example of the application of SES-based methodology to a significant real-world problem domain

### 15  PROCESSING NETWORKS AND PRAGMATIC FRAMES  273
Formulates networks of information processing nodes in terms of metadata operations and pragmatic frames

### 16  DYNAMIC PRAGMATICS: ISSUES AND METHODOLOGY  288
Discusses how to evaluate metadata in existing data exchange standards and to manage evolution over time

## PART IV  TESTING IN NET-CENTRIC ENVIRONMENTS

### 17  TESTING IN A NET-CENTRIC ENVIRONMENT: TECHNOLOGY BASIS  323
Automation based on the book’s concepts achieves rigor and faster test development in a net-centric environment
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 Testing in a Net-Centric Environment: Multiple Levels</td>
<td>361</td>
</tr>
<tr>
<td>How to use the framework to test at syntactic, semantic, and pragmatic levels</td>
<td></td>
</tr>
<tr>
<td>19 Bringing It All Together: Modeling and Simulation-Based Data Engineering</td>
<td>388</td>
</tr>
<tr>
<td>In-depth examination of the book’s contribution and its relation to the state-of-the-art</td>
<td></td>
</tr>
<tr>
<td>Glossary</td>
<td>418</td>
</tr>
<tr>
<td>Index</td>
<td>423</td>
</tr>
</tbody>
</table>

Online companion site: http://books.elsevier.com/companions/9780123725158
This book was born out of necessity of our work with various government agencies, testing facilities, standards organizations, and developers who are concerned with interoperability. While interoperability used to be a once-in-a-while issue, it is now becoming an institutional necessity with the planned transition of all Department of Defense operations to the Global Information Grid (GIG), a high-speed version of the World Wide Web. There are many compatibility issues because organizations have tended to develop their own unique approaches to representing data over the years. Now they will have to reconcile these differences so they can share data needed for effective collaboration. Unfortunately, there are few well-established methods for harmonizing data representations and testing their validity in real-world contexts. This book aims to fill some of the void.

The GIG is to be organized as a marketplace of open and discoverable web services called the Service Oriented Architecture (SOA), incorporating Semantic Web technologies as they mature. Indeed, the transition to the GIG/SOA is part of an accelerating trend to greatly increased complexity of information technology-
Based systems. And this complexity requires that interoperability testing become more rigorous, in-depth, and thorough. At the same time, to keep up with the rapid change and short development life cycles expected of web services, tests have to be ready to be conducted in much shorter time scales. This brings out the second motivation of the book: introducing model-based dynamic data engineering and simulation to automate testing methodology, thereby increasing rigor and speeding up test development in the emerging net-centric environment.

Where does the problem just outlined fit into the current array of concepts, methodologies, and technologies? Data must be encoded into a format to send it from a producer to a consumer. So there is a purely syntactic dimension to the problem: assure that the message formats are standardized and develop tests to see that the standards are adhered to. On the web, the eXtensible Markup Language (XML) is the standard format, and the problem can be viewed as assuring that all data sets are encoded into XML and that the standards for sending and receiving XML are adhered to. Unfortunately, the problem just starts at this level. There are myriad ways, or schemata, to encode data into XML and a good number of such schemata have already been developed. More often than not, they are different in detail when applied to the same domains. What explains this incompatibility?

There are various ways of breaking down a real situation and expressing it using a collection of terms. Two decompositions might well refer to slightly, or even markedly, different aspects of the same thing using the same terms. In other words, to interoperate there must be a common understanding of the semantics — what the data refer to. A similar problem in semantic web research has led to a focus on ontologies. These are logical languages that provide a common vocabulary of terms and axiomatic relations among them for a subject area. However, semantic web researchers typically seek to develop intelligent agents that can draw logical inferences from diverse, possibly contradictory, ontologies such as a web search might discover. In contrast, the newly emerging area of ontology integration assumes that human understanding and collaboration will not be replaced by intelligent agents. Therefore the goal is to create concepts and tools to help people develop practical solutions to incompatibility problems that impede “effective” exchange of data and ways of testing that such solutions have been correctly implemented.

But what does “effective” mean in the last sentence? A command is effective if it has the effect that the general intended it to have on his subordinates, which is to say, they carry out the given orders. So effective data exchange requires not only shared
agreement on syntax and on semantics, but also on pragmatics, the use to which data will be put.

Although the approach is generic, much of the application context of this book relates to geographic information as derived from intensive processing of terabytes of data collected from satellite and airborne sensors. This context is not only important on its own, but provides a microcosm for the larger problem domain of GIG/SOA interoperability. Geospatial sensors are sophisticated physics-based dynamic systems that are expensive to build and highly dependent on advanced engineering knowledge for their operation. This makes it nontrivial to retrofit them to the SOA concept of data-centered, interface-driven, loose coupling between producers and consumers. The SOA concept requires the development of platform-independent, community-accepted standards that allow raw data to be syntactically packaged into XML and accompanied by metadata that describes the semantic and pragmatic information needed to effectively process the data into increasingly higher-value products downstream.

This book presents an approach that was developed, in part, to meet the demands of developing interoperability standards and testing for their compliance in the geospatial sensor context. Given the physics-based nature of sensors it was natural to turn to a generic framework that had been developed to model and simulate such dynamic, continuous/discrete systems. The book presents a rigorous, yet intuitive, development of the System Entity Structure (SES) and its role as an ontology framework for static and dynamic world state descriptions, the latter expressed within the Discrete Event Systems Specification (DEVS) formalism.

From a static point of view, the SES is an ontology framework that is much closer to XML than that of semantic web ontology. At this level, the SES automates the creation of XML schemata using a data model that reflects system engineering concepts of hierarchical decomposition and specialization. In this sense, it occupies a space in the modeling pantheon similar to a commonly employed software engineering scheme, the Unified Modeling Language (UML). Among the advantages that it offers in this regard are a conceptual framework that is more expressive for the domain of real-world engineered systems and supporting set of tools for Schema development and testing in a virtual workspace at www.devsworld.org.

From a dynamic point of view, the SES together with the DEVS formalism offers a powerful system-theoretic framework for specifying families of dynamic services that can execute in simulated or real-time and interact with other services in a
net-centric environment. In this guise, the book provides an in-depth discussion of the automated development of test agent federations. These agents can validate the behavior, and evaluate the performance and effectiveness, of web-service collaborations through observation of their external message exchanges. As an example, we discuss end-to-end and multilevel testing of collaborating services on the GIG/SOA to achieve prespecified mission goals.