Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds

ONTOLOGIES FOR MODELING AND SIMULATION: ISSUES AND APPROACHES

Paul A. Fishwick

Computer and Information Science & Engineering
Bldg CSE, Room 301
University of Florida
Gainesville, FL 32611, U.S.A.

John A. Miller

Department of Computer Science
415 Boyd Graduate Studies Research Center
University of Georgia
Athens, GA 30602, U.S.A.

ABSTRACT

Ontologies represent the next important phase of the World Wide Web, creating a *semantic web* which links together disparate pieces of information and knowledge. Creating ontologies within computer simulation can be seen as a logical next phase of the web-based modeling and simulation thrust, where the emphasis is on knowledge and its representation rather than on run-time network characteristics. We introduce the concept of an ontology and then survey two groups performing research in this area at the Universities of Florida and Georgia, respectively.

1 INTRODUCTION

Web content has traditionally been created through the HyperText Markup Language (HTML). For example, most web pages, whether generated on the fly or created with a web page authoring tool, are structured in HTML. HTML tags define how text looks—the presentation of the document, but not its underlying semantic content. Even though presentation languages do provide a layer of semantics, the semantics are focused on whether to create a bold-faced font or how to indent a paragraph. The knowledge contained within the paragraph must be gleaned by the human reader of the web page, and so it becomes difficult to automate knowledge acquisition without a complex procedure involving natural language parsing. Suppose, for instance, that the web page had the following text phrase: "…the model contains a Markov chain of 4 states labeled M1, M2, M3 and M4…" It might be more productive to specify the structure of the Markov chain and then generate the text on the fly. That way, the knowledge takes center stage, and can be used to link to other knowledge about Markov chains, states, and models.

This more productive approach is basically the goal of the *Semantic Web* (Berners-Lee et al. 2001)—to surface the underlying knowledge about domains and objects, rather than focus solely on how this knowledge is presented in the way of text or diagrams.

An ontology represents a knowledge representation used to capture information and knowledge about a subject, generally within the structure of a semantic network, consisting of a diagram composed of nodes and arcs. An ontology may be expressed in the Extensible Markup Language (XML), the relatively new lingua franca of the web. Ontologies establish a semantic foundation for models, systems, results, applications and markup languages that enhances discovery, use and interoperation. The evolving web ontology initiative is developing a suite of languages to facilitate this: RDF, RDF-S, OWL, and SWRL.

Figure 1 illustrates a simple ontology, which relates various Petri Net (Fishwick 1995) models together.

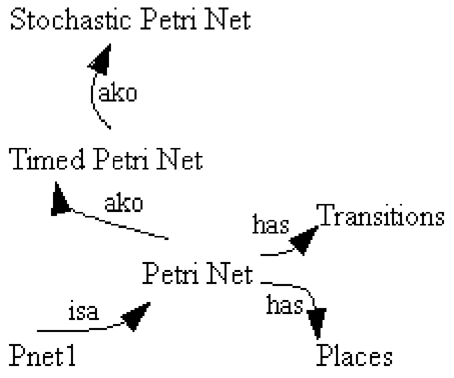

Figure 1: An Ontology for Petri Nets

It specifies the following knowledge: (1) A Petri net is a kind of (ako) Timed Petri Net, which in turn is a kind of Stochastic Petri Net; (2) Each Petri Net is composed of Places and Transitions; and (3) One particular Petri Net is labeled *Pnet1*. While simple in structure, this knowledge can be used in various ways. The ontology has archival or educational taxonomic properties, providing categorical information (i.e., that Timed Petri Nets are types of Stochastic Petri Nets). Also, the instance *Pnet1* defines one specific Petri Net that may be associated with a manufacturing process, for example. In this latter sense, the ontology not only contains information on categories, but also serves as a kind of “database” where instances can be stored and retrieved. In many implementations of ontologies and knowledge-bases, there is some difference noted between type (or class) information and instance information. For example, the types might create a schema definition, and the instances are stored in a tabular or database format. On the other hand, many of the examples of associate networks (Findler 1979, Sowa 2000) that can be said to have lead to the study of semantic networks and ontologies define semantic networks that are comprised of both classes and instances. The question is whether one considers the knowledge to consist only of the more abstract categories, or whether the objects are also considered an integral part of the ontology.

There are issues about ontology research which will need to be addressed within the simulation community. We mentioned one issue: *What are ontologies good for?* They can be used to store both class and instance information about models, model types, or model structures. But, perhaps the more important question is *how do we relate ontologies?* People will no doubt be creating different ontologies, requiring some form of standardization process to form communities intent on sharing the knowledge. Are there ways to relate ontological information automatically? For instance, if someone else created an ontology with the word *Petri Network* instead of *Petri Net* for Figure 1, can a program with natural language processing capabilities compare these two and form a synergy? Can ontologies be used as a basis for defining model repositories? It would appear that they can, as long as we allow instances and classes to be included.

Our purpose in this manuscript is not to answer all of these questions, but to begin the discussion of ontologies in modeling and simulation through example—briefly surveying work done in two different University settings: the University of Florida (UF) and the University of Georgia (UGA). The emphasis in each laboratory is somewhat different since at UF, the emphasis is on capturing mostly object or instance-based knowledge, whereas at UGA, the focus is more on the creation of a ontology for general stochastic models such as Markov Processes or Petri Nets.

2 UNIVERSITY OF FLORIDA

For the past four years, we have developed a software framework called RUBE. RUBE’s purpose is to provide a modeler with a way in which to better integrate the phenomenon being modeled and the model itself (Fishwick 2004). This integration is done using multiple visual modes of display (Hopkins and Fishwick 2003), allowing the dynamic models, as well as the phenomena, to be displayed in 3D. Figure 2 displays the RUBE architecture. Our use of ontologies within the RUBE project is founded on two approaches: (1) schema definitions and XML files for model types and model files; and (2) an OWL representation of a sample air reconnaissance scene. We proceed with these in sequence.

RUBE begins its process with two types of interfaces: a 2D interface using the SodiPodi tool, and a 3D one using Blender, which is a tool for authoring and animating 3D scenes. The simulation analyst builds a scene to be simulated, and then builds dynamic models of that scene. The dynamic models are translated into MXL (Multimodel eXchange Language) (Kim et al. 2002). MXL contains an ontology (or XML schema) defining certain model types and how they are defined. For example, a Finite State Machine contains an initial state, a set of states, and transitions. DXL (Dynamics eXchange Language) is a lower-level homogeneous block-model language capable of describing both synchronous and asynchronous execution of block networks. As such, DXL networks reflect behaviors such as those found in digital circuits as well as more loosely connected data flow networks. Both MXL and DXL are XML languages. Each has a schema, defining the language as an ontology.

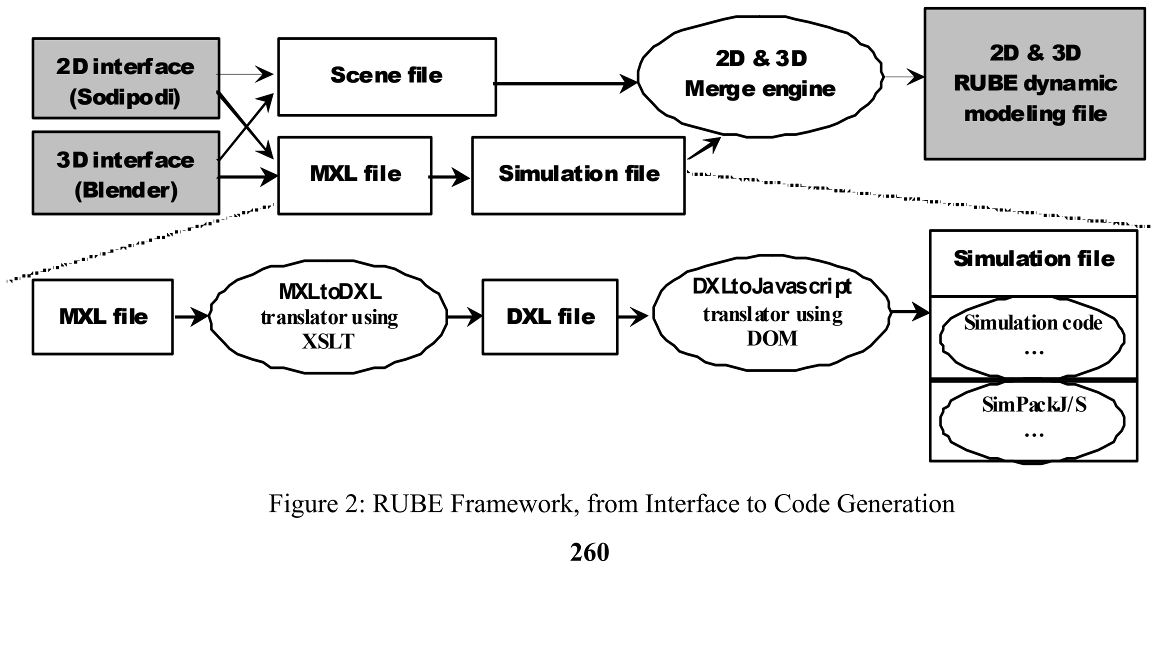

Figure 2: RUBE Framework, from Interface to Code Generation

Returning to the process defined in Figure 2, a model is converted into MXL and then DXL, and finally into a target language such as Java or JavaScript. This JavaScript code is then reinserted into the scene. This is done by first exporting the Blender scene into an X3D (eXtensible 3D) file and then defining the JavaScript in X3D script nodes. The final X3D scene file, intact with both geometric and dynamic properties, is then executed to yield the simulation. In addition to the work performed in RUBE, we have recently started to create an ontology that attempts to bring all knowledge about an application domain together. Figure 3 shows a network relating elements of a scene (JSTARS, F15, and UAV) with the geometry and dynamics of these objects; we have listed the defined styles that might be required during the formatting process. The idea here is that we can use RUBE to generate objects, which are then updated in the ontology. Moreover, we can add to the ontology manually, if necessary, to express the semantic relations. We are using OWL for expressing the network, and the Stanford Protégé tool (Protégé 2004) for managing the links. There is a two-way connection between Blender and Protégé so that information can be entered in Protégé and then appear in Blender, and vice versa.

3 UNIVERSITY OF GEORGIA

This section highlights ongoing work at UGA to develop a prototype ontology for discrete-event modeling and simulation. The prototype serves as a test bed for exploring issues in both ontology development and formal methods for simulation and modeling. Work on this prototype has brought to the forefront several issues

- Finding a suitable ontology language.
- Finding useful upper and mid-level ontologies.
- Creating a backbone taxonomy for the ontology..
- Choosing appropriate breadth and depth.
- Dealing with relationships to existing XML dialects.

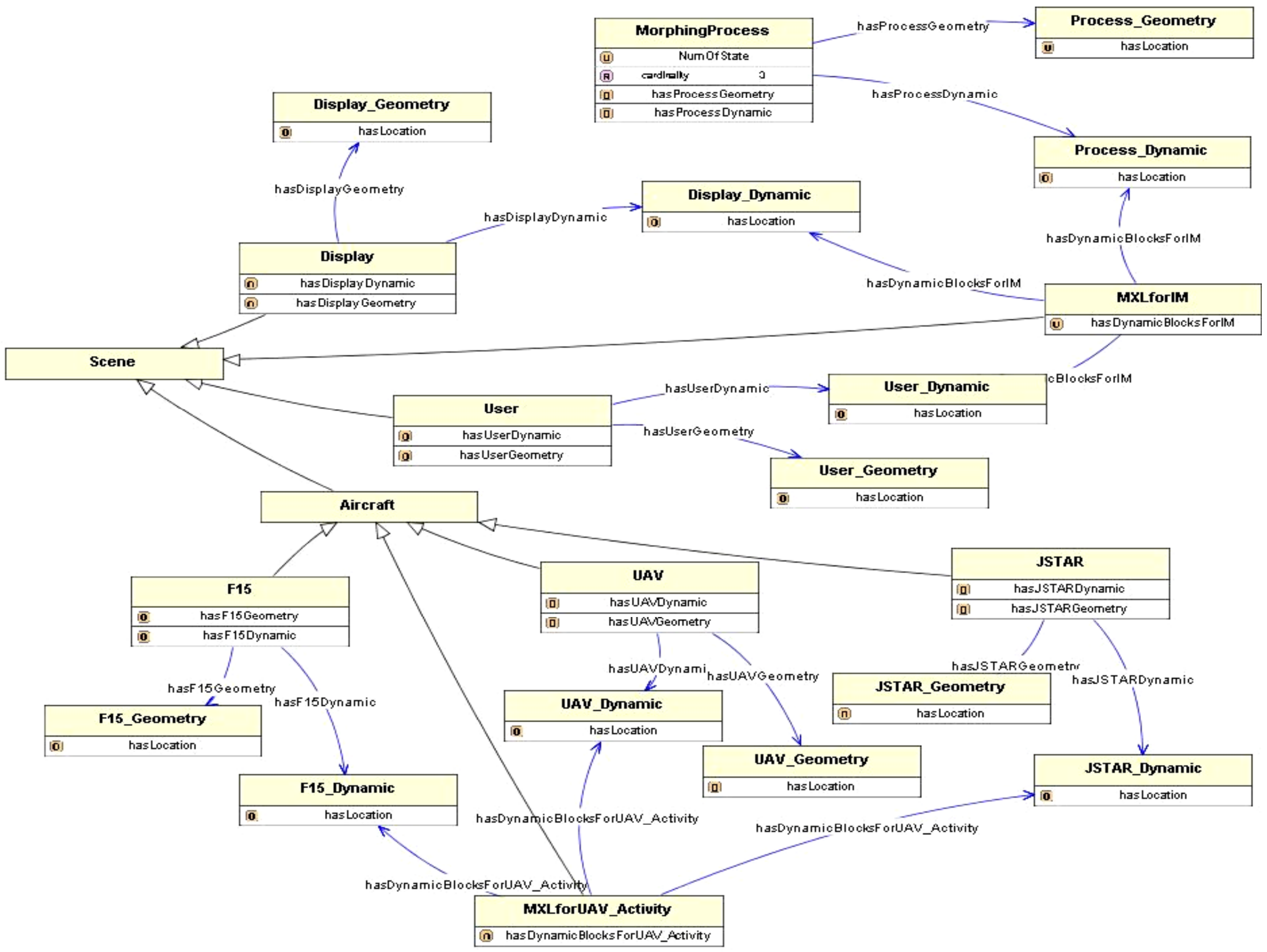

Figure 3: AN Ontology Defining an Air Battle Scene with Reconnaissance and Surveillance Aircraft

Although ontology languages such the Knowledge Interchange Format (KIF) were developed in the 1990's, the Semantic Web initiative has reinvigorated efforts to create a new ontology language which can have a wider appeal. KIF has the expressive power of First Order Logic (FOL) which enables complex concepts to be precisely described. The truth value of any FOL expression can be effectively checked (FOL is sound), but it is not possible to generate all true statements (FOL is incomplete). Taken together-FOL is said to be semi-decidable. For the Semantic Web, some researchers feel that a decidable language would be more appropriate, since limiting the languages expressivity will improve its computability/tractability. The Web Ontology Language (OWL) was designed with this expressivity/complexity trade-off in mind and comes in three flavors: OWL Lite, OWL DL and OWL Full, with the first two being decidable. Still some simple expressions such $x < y$ or $z = x + 1$ as well as recursive definitions are beyond the capabilities of OWL. To fill the gap, the Semantic Web Rule Language (SWRL) is being proposed. The current proposal for combining OWL with SWRL will lead to a semi-decidable language, although there is active research ongoing to find decidable subsets that are useful (Bechhofer et al. 2004). From our experience, complex ontologies suitable for modeling and simulation will need the capabilities of OWL+SWRL.

Higher level ontologies such as upper and mid-level ontologies are important for the following reason. A typical domain is likely to consist of upward of one hundred interrelated concepts. However, the concepts in a domain such as modeling and simulation will be defined using concepts from more general, foundational fields of study such as mathematics and statistics. If a modeling and simulation ontology is not built from these ontologies, then many concepts must be defined informally using natural language. Fortunately, work has begun on developing general upper ontologies as well as ontologies for mathematics and statistics. Two major initiatives for upper ontologies are the Suggested Upper Merged Ontology (SUMO) and the IEEE Standard Upper Ontology (SUO) (Niles and Pease 2001). In Mathematics, the Mathematics on the Net (MONET 2004) project is developing an OWL ontology based on the OpenMath (Caprotti et al. 2002) and MathML initiatives.

When first attempting to define an ontology in a new field, one may begin to feel as if they have chosen to swim across an ocean. The problem is so big and it is hard to know where to begin. We have cut down on the size of the problem by only considering discrete-event modeling and simulation. One way to begin is to review existing work on modeling and simulation taxonomies as well as related formal work. For example, several formal modeling techniques discuss homeomorphisms between their techniques and other well known techniques. Following this approach, one can identify several general types of discrete-event models such as Simulation Event Graphs (Yucesan and Schruben 1992), Generalized Semi-Markov Processes (Glynn 1983, Haas and Shedler 1987), Extended Stochastic Petri Nets (Dugan et al. 1984), and Process Templates (Cota and Sargent 1992). In (Miller et al. 2004) the DeMO ontology is discussed which uses these four types of models as roots of four model hierarchies, one for event, state, activity and process oriented models, respectively. Figure 4 is a screen-shot of part of the DeMO ontology (DeMO 2004) using the Jambalaya plug-in for Protégé.

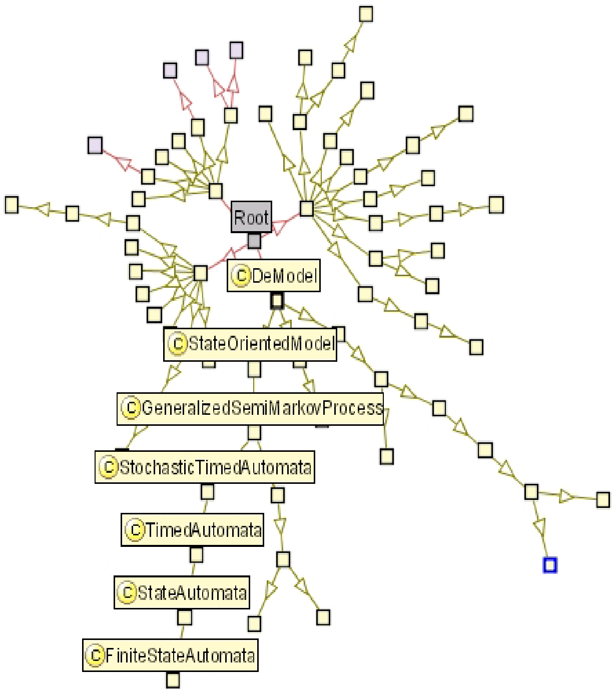

Figure 4: A Portion of the DeMO Ontology

The figure presents a radial view of the DeMO class hierarchy. The nodes represent ontology classes (concepts) and the edges symbolize a subclass relation. Other relations between different concepts are not shown in the picture. Clicking on the Finite-State Automata node produced the path from the root of the tree to the concept (i.e., Finite-State Automata is a subclass of State Automata, which is the subclass of Timed Automata).

This four way design begs the question of why four and why these four roots. Developing a more unified theory could be useful where each of the four roots in the DeMO ontology become a special case of one or more general models, so long as these more general models are not overly complex. Also, for clarity and historical preservation, using more than four roots may be a better approach. Ideally, over time a consensus on this issue should be developed.

Closely related to the issue of producing the backbone for the ontology are issues related to breadth and depth of the ontology. If an ontology is too broad and deep, it will be very hard to develop and maintain. At the other extreme, it will serve only a limited number of applications and might be more appropriately viewed as schema specification. The depth issue has important implications for automation. Many of today's ontologies will define concepts as classes with properties, where object properties imply a relationship. The definition of one concept is built from others. However, the definitions of many of the concepts are left to natural language. A good example in the DeMO ontology is the concept of probability. Defining probability formally would be a large undertaking, and better suited from some other community to do so.

The first part of the Semantic Web, the Extensible Markup Language (XML) has been well established for several years. XML is considered to be more semantically oriented than HTML since the tags are more meaningful, at least to humans and to applications that are coded with a given set of tags in mind. Because of this, much of what the OWL+SWRL ontologies will provide in the future is being attempted by defining dialects of XML such as the successful Mathematics Markup Language (MathML) and the Chemical Markup Language (CML) (Carlisle et al. 2003,**Error! Hyperlink reference not valid.**. These languages are formally specified using schema languages such as Document Type Definitions (DTD's) or XML Schema Definitions (XSD's). Similar efforts are ongoing in modeling and simulation. For example, the Simulation Reference Markup Language (SRML) (Reichenthal 2002) defines several tags/elements such as Script, Link, Item, EventClass, EventSink, Simulation, etc. SRML provides both DTD and XSD schemas and has been sent to the W3C for approval. Another example is the Extensible Modeling and Simulation Framework (XMSF), see (Brutzman 2004). These XML dialects are a useful source of information for the creation of modeling and simulation ontologies.

4 CONCLUSIONS

This paper overviews the semantic web technology and highlights ongoing work at the Universities of Florida and Georgia related to applying Semantic Web technologies to Modeling and Simulation (M&S). We have performed research the first level language of XML as well as some initial work at the higher levels of the Semantic Web (metadata with RDF and ontologies with OWL). These three levels allow for the creation of XML markup sub-languages for modeling and simulation which facilitate data exchange and interoperability, annotations of simulation resources available on the Web using RDF and finally, the creation of ontologies for modeling and simulation. When the language for the next level, SWRL providing rules, is completed, it will permit greater expressivity in ontologies as well as the development of rule oriented knowledge bases. In summary, the large-scale initiative to create the future Semantic Web, promises to open up and connect research and development groups on a Web scale and holds considerable promise for the M&S community.

There remain a number of issues of how the Semantic Web will affect modeling and simulation. It seems reasonable that, just as different groups have formed their own modeling communities, these same groups will formalize their model structures using XML, RDF, or OWL, with potential logical extensions with SWRL. Our community will require semi-automatic methods of comparing and contrasting, different Semantic model types, so that simply placing a modeling paradigm into XML serves as the first step in a more comprehensive process.

ACKNOWLEDGMENTS

The first author (Fishwick) would like to thank the National Science Foundation under grant EIA-0119532 and the Air Force Research Laboratory under grant F30602-01-1-05920119532. He would also like to thank Minho Park for his ongoing work relating to ontology construction as part of RUBE. The second author (Miller) would like to thank Gregory Baramidze for his work as the primary developer of the DeMO ontology.

REFERENCES

Bechhofer, S., I. Horrocks, and J. Pan 2004. *WonderWeb: Ontology Infrastructure for the Semantic Web.* Available online via `<wonderweb.man.ac.uk/deliverables/documents/D3.pdf>` [accessed August 20, 2004].

Berners-Lee, T., J. Hendler, and O. Lassila 2001. The Semantic Web, *Scientific American*, May 2001, pp. 35-43.

Brutzman, D. 2004. *Extensible Modeling and Simulation Framework (XMSF).* Available online via `<www.MovesInstitute.org/xmsf>` [accessed August 20, 2004].

Caprotti, O., D. Carlisle, and A. Cohen 2002. *The OpenMath Standard.* Available online via `<www.openmath.org/cocoon/openmath/standard/om11/omstd11.pdf>` [accessed August 20, 2004].

Carlisle, D., P. Ion, R. Miner and N. Poppelier, Eds, 2003. *Mathematical Markup Language (MathML) Version 2.0.* Available online via `<www.w3.org/TR/MathML>` [accessed August 20, 2004].

Cassandras, C.G. and S. Lafortune 1999. *Introduction to Discrete Event Systems*, Kluwer.

Cota, B. and R.E. Sargent. 1992. A modification of the process interaction world view. *ACM Transactions on Modeling and Computer Simulation (TOMACS),* 2 (2): 109 – 129.

DeMO 2004, Available online via `<chief.cs.uga.edu/~jam/jsim/DeMO>` [accessed August 20, 2004].

Dugan, J.B., K.S. Trivedi, R.M. Geist, and V. F. Nicola 1984. Extended Stochastic Petri Nets: Applications and Analysis. *Performance '84 - Models of Computer*

System Performance, E.Gelenbe, ed., Elsevier Sci. Publ. Co., Inc., pp. 507-519.

Fishwick, P.A. 1995, *Simulation Model Design and Execution: Building Digital Worlds*, Prentice Hall.

Fishwick, P.A. 2004, Toward An Integrative MultiModeling Interface: A Human-Computer Interface Approach to Interrelating Model Structures, *SCS Transactions on Modeling and Simulation*, to be published in the *Grand Challenges* special issue.

Findler, N. V., Ed, 1979. *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press.

Glynn, P. 1983. On the role of generalized semi-Markov processes in simulation output analysis. In *Proc. of the 1983 Winter Simulation Conference*. pp. 38--42.

Gruber, T.R. 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing, *Int. Journal of Human-Computer Studies*. Vol. 43, pp. 907-928.

Haas, P.J., and G.S. Shedler. 1987. Regenerative generalized semi-Markov processes. *Communications in Statistics-Stochastic Models* 3 (3): 409-438.

Hopkins, J. and P.A. Fishwick, P. A. 2003, Exploiting an Agent-Based Metaphor in Software Visualization using the rube System, *Journal of Visual Languages and Computing*, 14(1): 97-117.

Kim, T., J. Lee, and P.A. Fishwick 2002, A Two-Stage Modeling and Simulation Process for Web-Based Modeling and Simulation, *ACM Transactions on Modeling and Simulation*, 12(3): 230-248.

Miller, J.A., G. T. Baramidze, P.A. Fishwick, and A.P. Sheth 2004, Investigating Ontologies for Simulation Modeling, *Proceedings of the 37th Annual Simulation Symposium*, Arlington, VA, April 2004, pp. 55-71.

MONET Consortium. 2004. *The MONET Ontologies.* Available online via `<monet.nag.co.uk/cocoon/monet/publicdocs/monet_ontologies.html>` [accessed August 20, 2004].

Murray-Rust , P., and H.S. Rzepa 2001. *Chemical Markup Language. A Position Paper*. Available online via `<www.xml-cml.org/information/position.html>` [accessed August 20, 2004].

Niles, I., and A. Pease 2001. Towards a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, October 17-19, 2001.

Protégé Ontology Editor, 2004, Available online via `<protege.stanford.edu>` [accessed August 20, 2004].

Reichenthal, S.W. 2002. *SRML - Simulation Reference Markup Language.* Available online via `<www.w3.org/TR/SRML>` [accessed August 20, 2004].

Sowa, J.F. 2000 *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole, Pacific Grove, CA.

Yucesan, E. and L. Schruben 1992. Structural and behavioral equivalence of simulation models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 2 (1): 82 – 103.

AUTHOR BIOGRAPHIES

PAUL A. FISHWICK is Professor of Computer and Information Science and Engineering at the University of Florida. He pioneered or helped to initiate a number of areas within the modeling and simulation field, including multimodeling, fuzzy simulation, web-based modeling and simulation, and aesthetic computing. His first textbook *Simulation Model Design and Execution: Building Digital Worlds* was the first to ground the modeling problem of simulation within the context of programming language principles and types, and he has since published a 3-CD "gentle introduction" to modeling. Fishwick has been a Fellow of SCS since 1998, and serves on many archival journal editorial boards and advisory panels dedicated to modeling and simulation. In 1987, he founded the comp.simulation internet news group, and the electronic Simulation Digest. Dr. Fishwick was General Chair for the 2000 Winter Simulation Conference in Orlando. His main interest is in model representation. His e-mail address is `<fishwick@cise.ufl.edu>` and his home page is `<www.cise.ufl.edu/~fishwick>`.

JOHN A. MILLER is a Professor of Computer Science at the University of Georgia and has also been the Graduate Coordinator for the department for 9 years. His research interests include database systems, simulation and workflow as well as parallel and distributed systems. Dr. Miller received the B.S. degree in Applied Mathematics from Northwestern University in 1980 and the M.S. and Ph.D. in Information and Computer Science from the Georgia Institute of Technology in 1982 and 1986, respectively. During his undergraduate education, he worked as a programmer at the Princeton Plasma Physics Laboratory. Dr. Miller is the author of over 90 technical papers in the areas of database, simulation and workflow. He is an Associate Editor for ACM Transactions on Modeling and Computer Simulation and IEEE Transactions on Systems, Man and Cybernetics as well as a Guest Editor for the International Journal in Computer Simulation and IEEE Potentials.