Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

MULTIMODELS AND DYNAMIC STRUCTURE MODELS: AN INTEGRATION OF DSDE/DEVS AND OOPM

Fernando J. Barros

Departamento de Engenharia Informática
Universidade de Coimbra
P-3030 Coimbra, PORTUGAL

Bernard P. Zeigler

Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85712, USA

Paul A. Fishwick

Department of Computer & Information Science and Engineering
University of Florida
Gainesville, FL 32611-6120, USA

ABSTRACT

Constructing models of systems that change their structure over time has proved to be a challenging problem, with several proposed solutions. We present two of these approaches and discuss their integration. The Parallel Dynamic Structure Discrete Event System Specification is based on systems theoretic concepts and provides a formal specification of variable structure models. Object Oriented Physical Modeling (OOPM) extends the designs approaches of software engineering and employs both static and dynamic models to describe physical objects. OOPM is based on the multimodel concept to support adaptive structures. The integration is achieved by representing multimodels within the dynamic structure DEVS formalism. Since the latter has been implemented in HLA-compliant form, the integration has the practical consequence that multimodels can be directly mapped into distributed simulations.

1 INTRODUCTION

Dynamic structure models provide a framework for the representation of systems with evolvable structure. A comparison of several approaches can be found in Barros (1997a). A general theory of dynamic structure systems and also its application to networks of heterogeneous systems was developed by Barros (1997b). Multimodels were introduced by Ören (1991) and offer another possibility to represent dynamic structure models. In this paper we present the integration of the DSDE (Barros 1997a, 1998) and DEVS (Zeigler 1976) formalisms with the OOPM (Object Oriented Physical Modeler) environment Cubert et. al. (1996) and the multimodel concept (Fishwick 1995).

2 DISCRETE EVENT SYSTEM SPECIFICATION

The *Discrete Event System Specification* (DEVS) is a formalism introduced by Zeigler (1976) to describe discrete event systems. In the DEVS formalism a basic model is described by

$$M = (X, s_0, S, Y, \delta, \lambda, \tau)$$

where

- X is the set of input values
- S is the set of partial states
- s_0 is the initial partial state
- Y is the set of output values
- δ: $Q \times X^{\phi} \rightarrow S$ is the transition function, where
  - $X^{\phi} = X \cup \{\phi\}$
- $Q = \{(s,e) | \, s \in S, \, 0 \leq e \leq \tau(s)\}$ is the state set
- e is the time elapsed since last transition
- $q_0 = (s_0,0)$ is the initial state
- ϕ is the null value (absence of value)
- λ: $S \rightarrow Y$ is the partial output function
- τ: $S \rightarrow \mathbf{R}_0^+$ is the time advance function

If no event arrives to the system it will stay in partial state s for time $\tau(s)$ and when $e = \tau(s)$ the system changes to the state $(\delta(s,\tau(s),\phi),0)$. If an external event, $x \in X$, arrives when the system is in the state (s,e) the system will change to the state $(\delta(s,e,x),0)$. If an external event, $x \in X$, arrives when $e = \tau(s)$, the system changes to the state $(\delta(s,\tau(s),x),0)$. A complete description of DEVS semantics can be found in Zeigler (1976), Zeigler (1984), Zeigler, Kim and Praheofer (1998), Zeigler, Kim, and H. Praehofer (1997).

3 PARALLEL DYNAMIC STRUCTURE DISCRETE EVENT SYSTEM SPECIFICATION

The *Parallel Dynamic Structure Discrete Event System Specification* (DSDE) introduced by Barros (1997b), allows the specification of dynamic structure networks of discrete event systems (DEVS). A Parallel Dynamic Structure Discrete Event System Network is a 4-tuple

$$DSDEN_N = (X_N, Y_N, \chi, M_\chi)$$

where

N is the network name

X_N is the network input values set

Y_N is the network output values set

χ is the name of the dynamic network executive

M_χ is the model of the executive χ

The dynamic structure system network is defined with a special component, the *network executive* χ. The model of the executive, is a modified basic model and is defined by the 9-tuple

$$M_\chi = (X_\chi, S_\chi, s_{0,\chi}, Y_\chi, \gamma, \Sigma^*, \delta_\chi, \lambda_\chi, \tau_\chi)$$

where

$\gamma: S_\chi \rightarrow \Sigma^*$ is the structure function

Σ^* is the set of network structures

A structure $\Sigma_\alpha \in \Sigma^*$ associated with the executive partial state $s_{\alpha,\chi} \in S_\chi$, is given by

$$\Sigma_\alpha = \gamma(s_{\alpha,\chi}) = (D_\alpha, \{M_{i,\alpha}\}, \{I_{i,\alpha}\}, \{Z_{i,\alpha}\})$$

where

D_α is the set of component names associated with $s_{\alpha,\chi}$

for all $i \in D_\alpha$

$M_{i,\alpha}$ is the DEVS model of component i

for all $i \in D_\alpha \cup \{\chi, N\}$

$I_{i,\alpha}$ is set of components influencers of i

for all $i \in D_\alpha \cup \{\chi\}$

$Z_{i,\alpha}$ is the input function of component i

$Z_{N,\alpha}$ is the network output function

A detailed explanation of the DSDE formalism can be found in (Barros 1997c). The abstract simulators necessary to simulate DSDE models are described by Barros (1998).

4 OBJECT-ORIENTED PHYSICAL MODELING

OOPM extends the designs approaches of software engineering and uses both static and dynamic models to represent physical objects (Fishwick 1996). OOPM integrates models of physical systems with geometry and dynamics.

Model engineering is the process of building static and dynamic models for a physical scenario. The steps taken in this procedure are described in Figure 1.

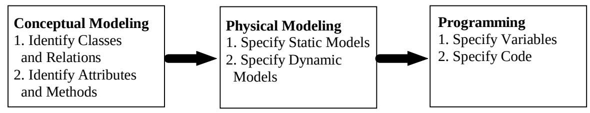

Figure 1:Model Engineering

The first phase is constructing a conceptual model of the physical scenario. To build such a model we must construct a class graph with relations among classes. Consider for example a car. We can create several types of cars: sport, luxury, sedans. Each of these are classes, and they are sub-classes of car since all of them are types of cars. This particular relation is called *generalization*. Another kind relation is aggregation since it involves a relation among a classes where there is a *part of* relationship.

The second phase is the specification of both static and dynamic models. For physical modeling the primary type of static model is the one that specifies the topology or geometry of a physical object. A dynamic model captures the way in which attributes change over time. Multimodels, to be presented in the next section, provides an approach to represent dynamic structure dynamic systems.

5 MULTIMODELS

Multimodels were introduced by Ören (1991) and provide a framework for representing models containing several submodels, where only one model can be active at any given time. Multimodels simulation environments and formal extensions have been described by Cubert, Goktekin and Fishwick (1997), Grossmann et. al. (1996) and Takahashi (1996). A multimodel $\mathcal{M}$ associated with models $M_1,\ldots,M_n$, can be represented by the System Entity Structure of Figure 2.

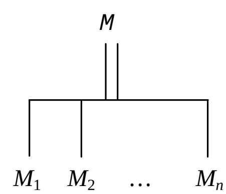

Figure 2: SES of a Multimodel

When certain conditions occur, the multimodel can change its structure by changing from the current model to another one. The same behavior can also be accomplished by the DSDE formalism. In this formalism a multimodel can be associated with a network model represented by the SES of Figure 3. The DSDE formalism demands for a separation of the changing model from the model responsible of handling the structural changes.

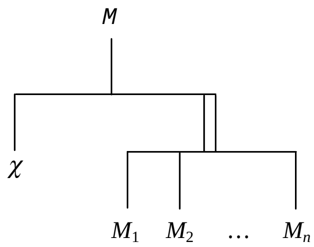

Figure 3: Multimodel Representation in the DSDE formalism

To show the behavior of the multimodel concept, we present, in Figure 4, the multimodel $\mathcal{M}$ in its initial structure where model M_0 is active.

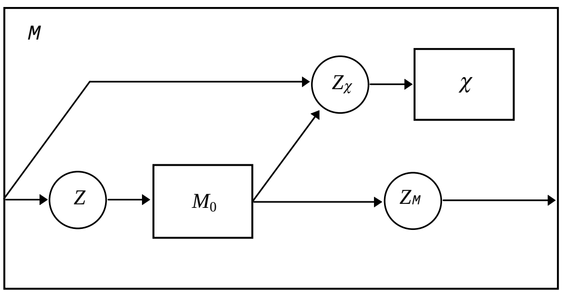

Figure 4: Multimodel in the Initial Structure

The executive when changes phase, also changes the current active model. In Figure 5 is represented the multimodel $\mathcal{M}$ when the active model is M_1.

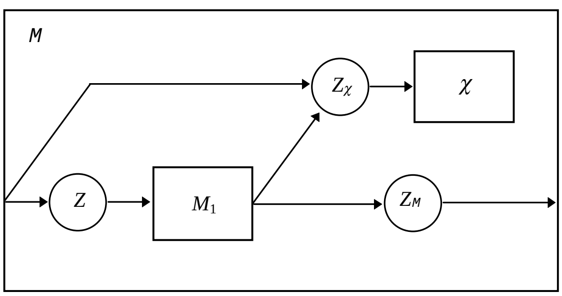

Figure 5: Change of Structure in the Multimodel

A multimodel M can be expressed in the DSDE formalism by the network

$$DSDEN_{\mathcal{M}} = (X_{\mathcal{M}}, Y_{\mathcal{M}}, \chi, M_\chi)$$

The model of the executive is given by

$$M_\chi = (X_\chi, \phi_0, \Phi, \gamma, \Sigma^*, \delta_\chi)$$

where

$\gamma: \Phi \rightarrow \Sigma^*$ is the structure function

Σ^* is the set of network structures

A structure $\Sigma_i \in \Sigma^*$ associated with the executive partial state $\phi_i \in \Phi$, is given by

$$\Sigma_i = \gamma(\phi_i) = (i, \{M_i\}, \{Z, Z_{\mathcal{M}}, Z_\chi\})$$

where

$\Phi = \{\phi_0, \phi_1, \ldots, \phi_n\}$

M_i is the model of component i

$Z: X_{\mathcal{M}} \rightarrow X$ is the input function of any component i

$Z_\chi: X_{\mathcal{M}} \times Y \rightarrow X_\chi$ is the executive input function

$Z_{\mathcal{M}}: Y \rightarrow Y_{\mathcal{M}}$ is the multimodel output function

We have omitted the set of influences from the original DSDE formalism for they are clear from input/output function definitions. All models M_i can be described by:

$$M_i = (X, Q, q_{0,i}, Y, f_i, \Lambda, \ldots)$$

This model is a continuous/discrete model and can be described by the DEVS&DESS formalism (Praehofer 1992). Combined models can produce the continuous value of the temperature and also discrete values indicating that state variables have reached some limit.

We consider that all the models share the same input X and output set Y, the same state set Q and the same output function Λ.

The multimodel formalism constraint the input functions Z and Z_χ, and the network output function $Z_{\mathcal{M}}$ to remain constant.

6 EXAMPLE: BOILING WATER SYSTEM

We describe now a multimodel application. We consider the pot of boiling water described by Fishwick and Zeigler (1992) and Fishwick (1995), and represented in Figure 6.

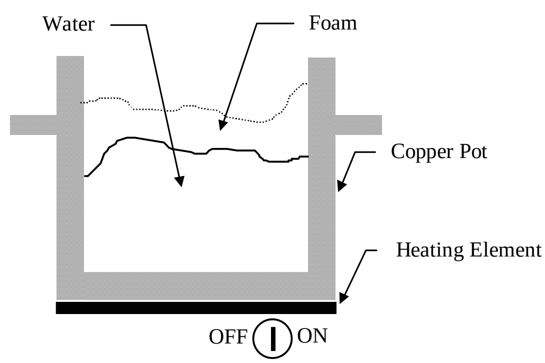

Figure 6: A Pot of Boiling Water

We focus only on the most important aspects of this multimodel system, and for simplicity we omit the aspects not directly connected with structural changes. The multimodel that represents the water boiler system is denoted $\mathcal{B}$ and can be expressed in the DSDE formalism by

$$DSDEN_{\mathcal{B}} = (X_{\mathcal{B}}, Y_{\mathcal{B}}, \chi, M_{\chi})$$

where

$X_{\mathcal{B}} = \mathbf{R} \times \{\text{none,ION,IOFF}\}$

$Y_{\mathcal{B}} = \mathbf{R}$

The multimodel receives the value of the pot temperature and yields the water temperature. The model of the executive is given by

$$M_{\chi} = (X_{\chi}, \phi_0, \Phi, \gamma, \Sigma^*, \delta_{\chi})$$

where

$X_{\chi} = \{\text{ION,IOFF,T100,Ta,HfHt,Hw0}\}$

$\Phi = \{\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$

The executive receives signals from both the pot model and from the outside world. Outside signals include information about the status of the heating element (ON/OFF). The pot model sends information about temperature. For each model $M_i = \{X, Y, Q, q_{0,i}, f_i, \Lambda, \ldots\}$ we have

$X = \{\}$

$Y = \mathbf{R} \times \{\text{none,T100,Ta,HfHp,Hw0}\}$

The multimodel has 6 models $M_0,\ldots,M_6$, corresponding to each executive phase:

M_0 (Cold): $f_0(T,Hw,Hf) = (\alpha,0,0)$

M_1 (Heating): $f_1(T,Hw,Hf) = (k_1(Tp - T),0,0)$

M_2 (Cooling): $f_2(T,Hw,Hf) = (k_2(\alpha - T),0,-k_3)$

M_3 (Boiling): $f_3(T,Hw,Hf) = (0,-k_4,k_5)$

M_4 (Overflow): $f_4(T,Hw,Hf) = (0,-k_4,0)$

M_5 (Underflow): $f_5(T,Hw,Hf) = (0,0,0)$

where variables have the following meaning:

T: temperature of the water.

Tp: temperature of the pot.

Hw: height of the water.

Hf: height of the foam.

Hp: height of the pot.

The translation from inputs to conditions is given by:

ION: I = ON.

IOFF: I = OFF.

T100: $T = 100$.

Ta: $T = \alpha$.

HfHp: $Hf = Hp$.

Hw0: $Hw = 0$.

The executive phase transition is represented in Figure 7. Its starts in phase ϕ_0 (Cold) which is associated with model M_0. When it receives the ON command it changes to phase ϕ_1 (Heating). In this phase temperature rises and eventually it will reach 100 degrees; at this moment, the current model M_1 sends a T100 signal indicating the water has started boiling. The executive upon the receipt of this signal changes its phase to Boiling and changes model to M_3.

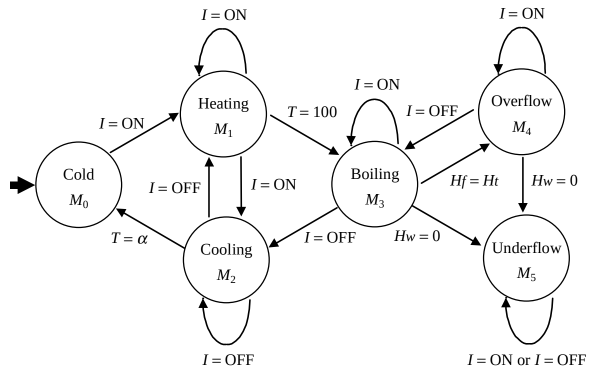

Figure 7: Automaton for the Network Executive

Change from a model to a new one is instantaneous and it is assumed that the new model will start in the state the removed model was in.

7 BOILING WATER REVISITED

We describe a solution in the realm of the DSDE formalism that explores a fine-grained version of the previous models. In this solution we use a model for each variable of the original model. The changes in structure can occur now in each one of the models. For temperature *T* we have:

$M_{0,T}$: $f_{1,T}(T) = \alpha$

$M_{1,T}$: $f_1(T) = k_1(Tp - T)$

$M_{2,T}$: $f_{2,T}(T) = k_2(\alpha - T)$

$M_{3,T}$: $f_{4,T}(T) = 0$

for *Hw* we have:

$M_{0,w}$: $f_{0,w}(Hw) = 0$

$M_{1,w}$: $f_{1,w}(Hw) = -k_4$

and for *Hf* there are 3 different models:

$M_{0,f}$: $f_{0,f}(Hf) = 0$

$M_{1,f}$: $f_{1,f}(Hf) = -k_3$

$M_{3,f}$: $f_{2,f}(Hf) = k_5$

The SES of the overall model is represented in Figure 8. We can obtain a model by a combination of specializations M_T, M_w and M_f. When, for example, the phase changes from Cold to Heating, the only model that needs to be changed is M_T from $M_{0,T}$ to $M_{1,T}$, as depicted in Figure 9. Also $M_{3,T}$, $M_{0,w}$ and $M_{0,f}$ are equal, which can provide model reuse.

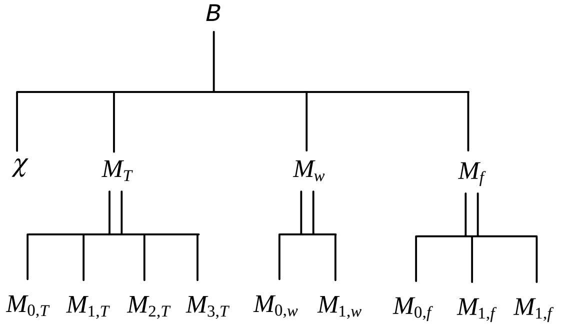

Figure 8: SES for Fine-Grained Solution

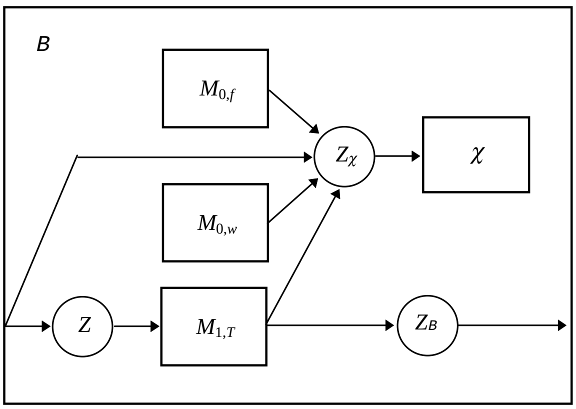

Figure 9: Model in the Heating Phase

The executive and the structure functions need to be changed. Namely, the structure function does not map an executive phase to a single model. The executive must receive now both *Hf* and *Hw* variables and make their comparison to change phase. Thus the executive itself needs to be modeled as an hybrid model.

8 DEVS/HLA

HLA (High Level Architecture) is a standard of the Department of Defense for simulator interoperability and reuse to be adopted by all defense contractors and agencies by 2001. As illustrated in Figure 10, the operational form of the standard is a Run Time Infrastructure (RTI) consisting of a centralized executive and node ambassadors (software equivalents of network interface cards) that support communication among simulations, called federates. DMSO (Defense Modeling and Simulation Office) has developed an RTI in C++ (currently version 1.3) for use in the public domain. HLA supports a number of features including establishing, joining and quitting federations, time management and inter-federate communication.

DEVS/HLA is an HLA-compliant modeling and simulation environment formed by mapping the DEVS-C++ system (Zeigler et. al 1997) to the C++ version of the DMSO RTI. While HLA supports interoperation at the simulation level, DEVS/HLA supports modeling level features inherited from the DEVS – as a generic dynamic systems formalism, with a well defined concept of coupling of components, hierarchical, modular construction, support for discrete event approximation of continuous systems and an object-oriented substrate supporting repository reuse (Zeigler and Lee 1998). DEVS/HLA supports high level model building in DEVS terms, shielding the modeler from the underlying programming details that have to be managed to establish and participate in an HLA federation.

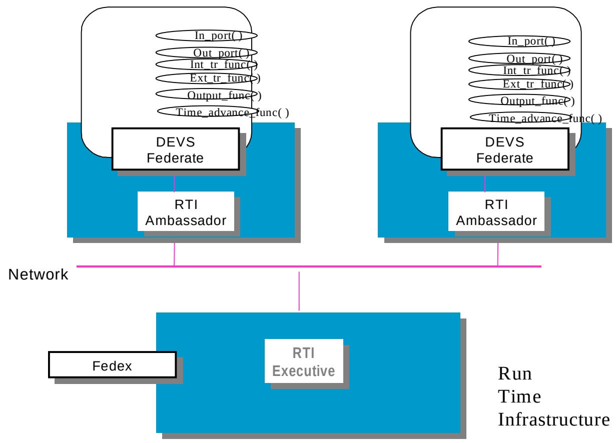

Figure 10: DEVS/HLA

DEVS/HLA federates communicate through the standard DEVS interfaces of the DEVS models they contain (Figure 10). In this manner, DEVS/HLA can supply an HLA-compliant layer to yet higher level modeling formalisms such as OOPM to be described below. The key requirement is that such formalisms support a concept of modularity in which component objects are coupled together through distinguishable input/output interfaces. DEVS atomic models can supply a wrapper to such components (which become federates) by having the DEVS transition, output and time advance functions invoke corresponding methods within the modules. Similarly, the manner in which outputs are routed to inputs in the higher level formalism must be expressed as couplings among ports in DEVS terms. This will allow messages to be sent from one federate to another using the underlying HLA message transmission facilities called "interactions".

The resulting high level environment, in the case of OOPM, is depicted in Figure 11. Models developed in OOPM can be automatically translated into DEVS-C++ code (once the required mappings are defined) and then executed in the DEVS/HLA environment over any TCP/IP network of hosts executing the HLA C++ RTI. Further to obtain the dynamic structure capabilities required by DSDE, we can exploit HLA support for federates to join and quit a federation while it is executing.

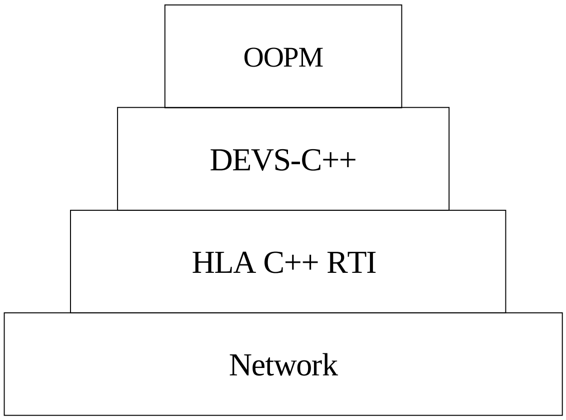

Figure 11: HLA/DEVS Supporting OOPM

9 CONCLUSIONS AND FUTURE WORK

We have discussed the integration of the dynamic structure DEVS formalism with the multimodel paradigm. The integration was achieved by representing multimodels within the DSDE and DEVS formalisms. The integration provides the basis for mapping OOPM and its multimodels directly into DEVS/HLA distributed simulations. Further work is needed to verify the mappings, implement them in software, and experiment with the resulting environment.

REFERENCES

Barros, F. J. 1995. Dynamic Structure Discrete Event System Specification: A New Formalism for Dynamic Structure Modeling and Simulation. In *Proceedings of the 1995 Winter Simulation Conference*, 781-785.

Barros, F. J. 1996. Dynamic Structure Discrete Event System Specification: Formalism, Abstract Simulators and Applications. *Transactions of the Society for Computer Simulation* 13(1): 35-46.

Barros, F. J. 1997a. Dynamic Structure Discrete Event Systems: A Comparison of Methodologies and Environments. In *Proceedings of SPIE 11th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls: Enabling Technology for Simulation Science* 3083: 268-277.

Barros, F. J. 1997b. Modeling Formalisms for Dynamic Structure Systems. *ACM Transactions on Modeling and Computer Simulation*, Vol. 7, No. 4, 501-515.

Barros, F. J. 1998. Abstract Simulators for the DSDE Formalism. In *Proceedings of the 1998 Winter Simulation Conference*. To be published.

Cubert, R., T. Goktekin, and P. A. Fishwick. 1997. MOOSE: Architecture of an Object-Oriented Multimodeling Simulation System. *Proceedings of SPIE 11th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls: Enabling Technology for Simulation Science* 3083: 78-88.

Fishwick, P. A. 1995. *Simulation Model Design and Execution*. Englewood Cliffs: Prentice-Hall.

Fishwick, P. A. 1996. Extending Object-Oriented Design for Physical Modeling. University of Florida/ CISE Technical Report. http://www.cise.ufl.edu/~fishwick/tr/tr-026.html.

Fishwick, P. A. 1998. An Architectural Design for Digital Objects. In *Proceedings of the 1998 Winter Simulation Conference*. To be published.

Grossmann, W., H. Werthner, and M. Hitz. 1996. An Object-Oriented Simulation Environment Supporting Self-Adapting Model Structure and Qualitative Models. *Proceedings of the 6th AI Conference on AI, Simulation and Planning in High Autonomy Systems Winter Simulation Conference*, 125-132.

Ören, T. I. 1991. Dynamic Templates and Semantic Rules for Simulation Advisors and Certifiers. In *Knowledge-Based Simulation: Methodology and Application*, ed. P. A. Fishwick and R. B. Modjeski (eds.), 53-76. New York: Springer Verlag.

Praehofer, H. 1991. *Systems Theoretic Foundations for Combined Discrete-Continuous System Simulation*. Ph.D. Dissertation, University of Linz.

Takahashi, S. 1996. General Morphism Concept for Multimodeling. In *Proceedings of the 6th AI Conference on AI, Simulation and Planning in High Autonomy Systems Winter Simulation Conference*, 275-280.

Zeigler, B. P. 1976. *Theory of Modelling and Simulation*. New York: Wiley.

Zeigler, B. P. 1984. *Multifaceted Modelling and Discrete Event Simulation*. London: Academic Press.

Zeigler, B. P., and H. Praehofer. 1989. Systems Theory Challenges in the Simulation of Variable Structure and Intelligent Systems. In *CAST-Computer-Aided Systems Theory*, ed. F. Pichler and Moreno-Diaz, 41-51. Berlin: Springer Verlag.

Zeigler, B. P., D. Kim, and H. Praehofer. 1997. DEVS Formalism as a Framework for Advanced Distributed Simulation. In *Proceedings of the First International Workshop on Distributed Interactive Simulation and Real Time Applications* (in conjunction with *MASCOTS'97 - International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*). Eilat, Israel: IEEE Press, San Diego, CA.

Zeigler, B. P., et al., 1997. The DEVS Environment for High-Performance Modeling and Simulation. *IEEE C S & E* 4(3): 61-71.

Zeigler, B. P. and J. S. Lee. 1998. Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment. in Enabling Technology for Simulation Science(II), SPIE AeoroSense 98. Orlando, FL.

Zeigler, B. P., T. G. Kim, and H. Praehofer. 1998. *Theory of Modeling and Simulation*. 2 ed., New York, NY: Academic Press.

AUTHOR BIOGRAPHIES

FERNANDO J. BARROS is an assistant professor at the Department of Informatics of the University of Coimbra. He received M.S. and Ph.D. degrees from the University of Coimbra. He is a member of SCS and ACM. His research interests include Theory of Modeling and Simulation and adaptive systems.

BERNARD P. ZEIGLER is Professor of Electrical and Computer Engineering at the University of Arizona, Tucson. He received his B. Eng. Phys from McGill University, 1962, M.S.E.E. from MIT, 1964, and Ph.D. from the University of Michigan in 1968. He was elected as Fellow of the IEEE for his innovative work in discrete event modeling theory. In 1995, Zeigler served on a National Research Council committee to suggest directions for information technology in the 21st Century US Army and is currently a member of an NRC committee given a similar task by the US Navy, focusing on modelling and simulation. Zeigler is currently editor-in-chief of the Transactions of The Society for Computer Simulation.

PAUL A. FISHWICK is a Professor of Computer and Information Science and Engineering at the University of Florida. He received the BS in Mathematics from the Pennsylvania State University, M.S. in Applied Science from the College of William and Mary, and Ph.D. in Computer and Information Science from the University of Pennsylvania in 1986. He his a fellow of the Society of Computer Simulation, and a Senior member of the IEEE. Dr. Fishwick was chairman of the IEEE Computer Society technical committee on simulation (TC-SIM) for two years (1988-90) and he is on the editorial boards of several journals including the *ACM Transactions on Modeling and Computer Simulation*, *IEEE Transactions on Systems, Man and Cybernetics*, *The Transactions of the Society for Computer Simulation*, *International Journal of Computer Simulation*, and the *Journal of Systems Engineering*.