Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

DEVS FORMALISM AND METHODOLOGY: UNITY OF CONCEPTION/DIVERSITY OF APPLICATION

Bernard P. Zeigler
Sankait Vahie

AI & Simulation Research Group
Dept. of Electrical & Computer Engineering,
University of Arizona,
Tucson, Az. 85721.

ABSTRACT

DEVS (Discrete Event System Specification) is a general modelling formalism with sound semantics founded on a system theoretic basis. This gives it a claim to be universal for formalisms describing Discrete Event Dynamic Systems (DEDS). This means that any other formalism such as Petri nets, which have become very popular for DEDS control can be embedded in it. Moreover, DEVS extends to the continuous case thus facilitating combined discrete/continuous modelling. The universality of DEVS is significant because it has been implemented in a variety of simulation environments, as extensions of diverse underlying Object-Oriented languages such as CLOS and C++. This gives it not only the power of formal rigor but also the practical capability of application to real world complex systems. The DEVS formalism has associated with it a characteristic abstract simulation engine architecture that can be realized in diverse sequential and parallel/distributed platforms. It is especially suitable for the simulation study of complex technical and natural systems with intelligent components and for long term model development capability acquired through systematic reusability.

This tutorial will present the basic concepts of the DEVS formalism and its associated simulation methodology. These concepts will be illustrated in the context of large scale ecosystems modelling.

1 INTRODUCTION

Ho (1989) lists as challenges to a universal modelling framework for Discrete Event Dynamic Systems (DEDS) the need to address issues under the following headings:

- Discontinuous nature of discrete events
- Continuous nature of most performance measures
- Importance of probabilistic formulation
- Need for hierarchical analysis
- Presence of dynamics
- Feasibility of the computational burden.

Although many formalisms have been developed that partially address these challenges, there is no single formulation that meets them all. However, a system theoretic framework can provide a sound modelling foundation that can address these issues in a unified manner. In the early seventies, such a systems theory framework was developed for computer-based modelling and simulation (Zeigler 1976). The common mathematical means of system descriptions, using difference and differential equations, were characterized formally as system specifications. In other words, they could be viewed as shorthand means of providing the data necessary to uniquely specify a system in the class of system models. Hence in Figure 2, discrete time system specifications (DTSS) and differential equation system specifications (DESS) form subclasses of the universal class of systems.

In contrast, the form of system description intrinsic to the computer-based discrete event simulation languages was not mathematically well characterized. Code implementation details obscured the formal core of such models and it was often tacitly assumed that no useful code-independent model description could be given. Thus adopting the system formalism perspective facilitated newer forms of expressions that could subsume existing ill-formalized modelling approaches of practical importance. This perspective suggested that discrete event simulation models be captured as a subclass of systems in the form of DEVS (Discrete Event System Specification). Because of its system theoretic basis, DEVS stakes a

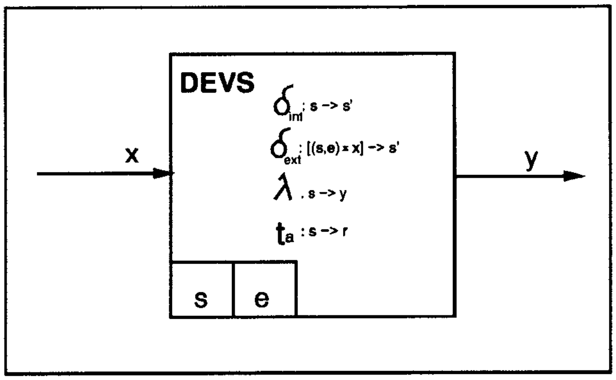

Figure 1: Discrete Event System Specification

claim to be the most inclusive formalism for DEDS. Indeed, DEVS is exactly the short-hand needed to specify systems whose input, state and output trajectories are piecewise constant. The step-like transitions are identified as discrete events. For example, a change in the input trajectory can be spoken of as the arrival of an external event or input. DEVS views a system state as a pair (s, e) where s is a so-called sequential state and e is the (elapsed) time that the system has been in state s since it last entered it (figure 1). A time-advance function, ta, is part of the specification — a distinguishing feature of DEVS, which assigns to every sequential state a maximum residence time. Therefore, only pairs (s, e) where $0 \leq e \leq ta(s)$ are system states. An external input arriving when the system is in state (s, e) causes a transition to a new state of the form $(s', 0)$ where s' can depend on (s, e) and the input. Outputs are generated just before internal transitions.

Embedded in systems theory, DEVS supports an open approach to formalism development allowing the researcher to explore new formalisms and to characterize their expressive power. Indeed, one can ask what is the largest class of systems that can be represented by DEVS (figure 2). It turns out that this, DEVS-representable-system class, contains systems whose input and output are event-like in nature (the input and output trajectories are all piecewise constant) but whose state behavior can take on any form, (e.g., a DESS) that is compatible with the interface (input-output) restrictions (Zeigler, 1990) (figure 3). On the other hand, DEVS itself can be embedded in larger system classes. The DEV & DESS formalism (Praehofer, 1991) combines both DEVS and DESS (i.e, discrete event and differential equation) expressibility (figure 2).

The DEVS formalism is also suited for design of simulation engine architectures. The correctness of existing discrete event simulation algorithms can be investigated in a general formal framework. More significantly, this framework can be used for developing and verifying new approaches especially in parallel and distributed architectures. Because the formalism was developed from a general system theory basis it is, in principle at least, unbiased by the sequential implementations which were the first to achieve wide spread use.

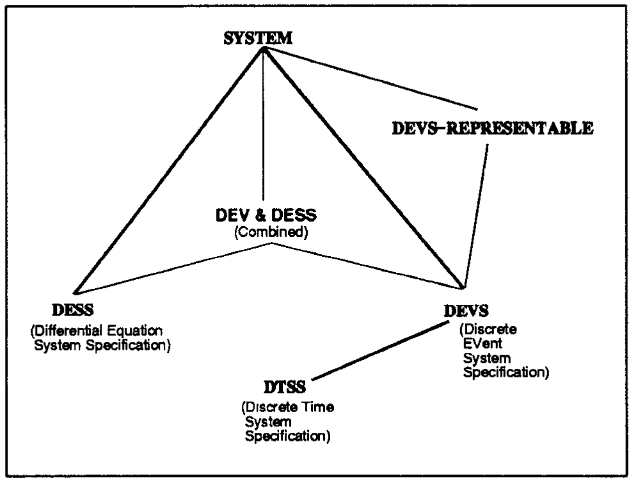

Figure 2: System Classes & Formalism

Developments in DEVS and related system formalisms, show that expressibility and simulation architecture designs are not just exercises in mathematical virtuosity. They lead to improved concepts and software that render modelling more expressive and simulation more efficient and usable.

In the rest of this tutorial, we present an example that illustrates the utility of the DEVS methodology. We show how DEVS provides a cellular space framework that can be integrated with terrestrial databases to enable modelling and simulation of large scale landscape ecosystems.

2 DEVS EXAMPLE: LANDSCAPE ECOSYSTEM MODELLING AND SIMULATION

Concern for the global environment has engendered much interest in monitoring and predicting landscape and ecosystem changes for large geographic regions. Satellite and land-based earth observation programs to gather and archive large masses of data have in turn stimulated research on data analysis and modeling methods to exploit this data for prediction and control. The task of simulation software is to pro-

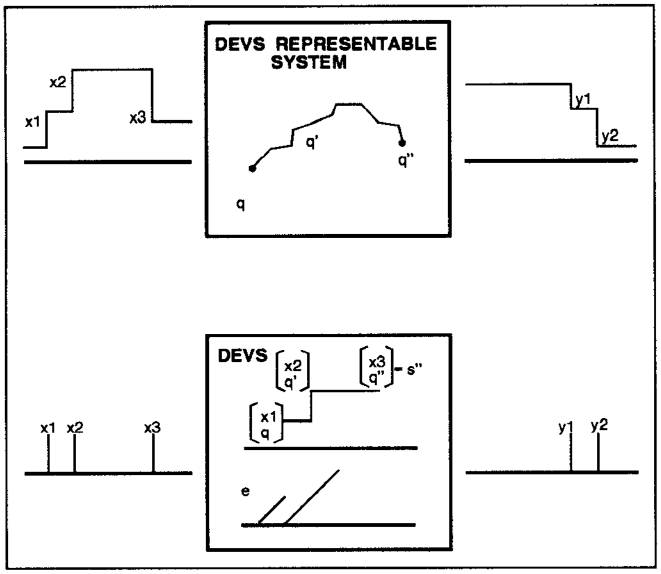

Figure 3: DEVS Representation of Systems

vide the dynamic state projection necessary for understanding complex system behavior and ultimately providing decision support for ecosystem management. However, state-of-the-art simulation software does not have adequate capability to access the huge volumes of data necessary for realistic landscape representation. Geographical Information System (GIS) technology provides powerful data bases for storing and retrieving spatially referenced data. Spatial information is stored in many different themes representing quantitative, qualitative, or logical information. These data can have different resolutions that range from detailed local information to small scale satellite imagery. GIS operators provide the means for manipulating and analyzing layers of spatial information and for generating new layers. Since it allows distributed parameterization, a GIS is useful for ecological models that need to explicitly incorporate the spatial structure and the variability of system behavior (Band and Wood 1988; Running et al. 1989; Band et al. 1991). A raster-based GIS represents spatial information as a grid of cells, and each cell corresponds to a uniform parcel of the landscape. Cells are spatially located by row and column and the cell size depends on the resolution required.

Interest has been expressed in using GIS for simulation of spatial dynamic ecological processes (Berry 1987; Itami 1988; Green et al. 1989; Costanza et al. 1990; Baumans and Sklar 1990; Sklar and Costanza 1991; Vasconcelos and Guertin 1992; Ball and Guertin 1992). However, GIS systems do not include procedures for handling time, they are designed to process entire arrays of data, and cannot easily address varying localized operations across the spatial grid (Berry 1987; Running et al. 1989; Band et al. 1991; Vasconcelos and Guertin 1992)

Landscape ecosystem models employ localized neighborhood computations, such as diffusion processes that involve movement through space, (e.g., oil spills, seed dispersal, fire spread, insect infestation). These are usually represented as partial differential equations (PDEs) that have to be discretized in the form of finite differences or finite elements (Fahrig 1988; Band and Wood 1988). Such representations assume continuity of space and time and so are too fine-grained to enable feasible simulations of large regions. For example, areas requiring a landscape-scale grid matrix of 500X500 cells are computationally impractical. As an alternative, cellular automata (CA) have also been used for implementing spatial dynamic ecological models in cellular spaces (Hogeweg 1988, Green et al. 1989, Itami 1988). They can be seen as discrete models of spatio-temporal dynamics obeying local laws. Not only does the CA methodology incorporate discretized PDEs, it also, by employing more qualitative state representations, makes it easier to express the dynamics of interacting discrete units in space.

Nevertheless, the coastal landscape models developed by Sklar et al. (1985) and Costanza et al. (1990) make clear the limitations of the CA framework. Its homogeneous cellular structure and synchronous time advancement are too rigid to easily accommodate the diversity of processes that interact at overall ecosystem scale. A generalization of the CA methodology, viz., discrete-event object-oriented simulation of hierarchical, modular models, has been shown to offer the flexibility to include the full spectrum of process description levels ranging from individuals to aggregated populations (Savatsky and Reynolds 1989; Rozenblit and Jankinski 1990; Baveco and Lingeman 1992). As earlier suggested by Couclelis (1985), the utility of linking the DEVS formalism for cellular spaces (Zeigler, 1984) with GIS was recently demonstrated by Vasconcelos and Zeigler (1993) and Vasconcelos et al. (1993). The DEVS-Scheme knowledge-based simulation environment (Zeigler 1990) was the basis for this work.

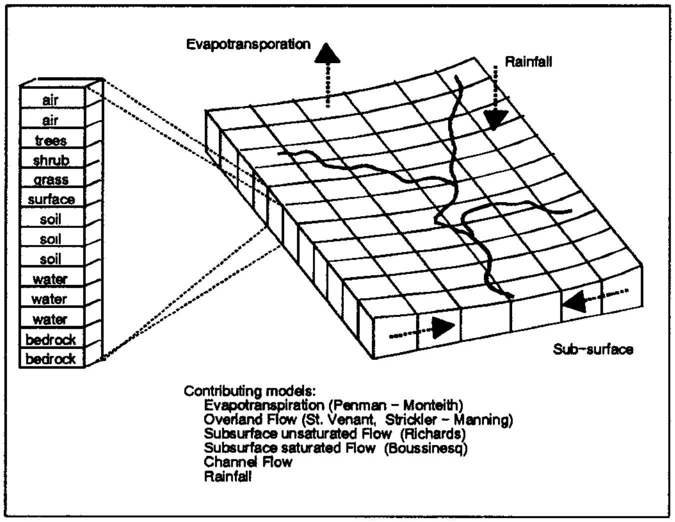

Figure 4: DEVS Cellular Space Representation of Watershed

3 CELLULAR AUTOMATON FORMULATION OF LARGE SCALE ECOSYSTEM MODELS

To illustrate these ideas, we consider a watershed with simplified hydrology as an example of landscape level modelling. There have been several efforts directed at developing physically based distributed hydrologic models at the catchment scale However, to date, no modeling efforts have aimed at providing a spatially distributed visualization of processes and interactions controlling water movement at a catchment scale. The DEVS-based model below can provide such visualization.

The watershed is modelled as a bounded 3-D grid (figure 4), with each cell representing a cube whose dimensions are determined by the chosen resolution. A column of cells will typically start with bedrock at the bottom, move up through several layers of soil with differing hydrologic parameters, reaching a surface layer, and continuing on to several cells of air. Rain is represented as inputs to the top boundary layer of cells and will infiltrate downwards and sideways to the watershed basin. Each cell is represented by a local component model and the cellular heterogeneity requires that each such model have parameters specifying its unique air, soil, or bedrock characteristics. This information is stored in the GIS and downloaded into each model initially and at experimental-frame specified points in the simulation run.

Existing hydrologic process models (Freeze and Herlan 1969; Woolhiser 1973; Morris, 1980; Alonso and De Coursey, 1985; Abbott et al. 1986; Bathurst, 1986) were employed to develop the cell model. It comprises two major components: hillslope and channel. The hillslope component is made up of three storage systems: surface storage, soil storage, and ground water storage. The primary components of the land phase of the hydrologic cycle are evapotranspiration, infiltration, unsaturated and saturated subsurface flow.

In the standard model-to-simulator mapping (Zeigler, 1984), each cell model is assigned to a simulator with interprocessor communication in a 3-dimensional nearest-neighbor grid. An execution cycle has two phases:

Concurrently, each simulator:

1. Gathers neighbor inputs,
2. Computes local model state transition, and returns to phase 1.

In the watershed case, the neighbor inputs are excess water amounts representing lateral or vertical flow. The local state transition is an integration step of the differential equation-based hydrologic model.

4 DEVS CELLULAR SPACE FORMULATION OF WATERSHED MODEL

The watershed model exhibits event-like behavior arising from water retention in the cells. Due to the storages, each cell can be viewed as a reservoir which does not transmit water until its capacity is reached. This means that intercell data exchange need only occur at discrete instants rather than at each cycle as above. DEVS cellular space representation affords effective means to represent continuous transition/discrete interaction behavior as well as to accommodate simulation engines that can exploit these representations to achieve greater execution speed.

To sketch the basic idea for DEVS cellular space representation, we let each cell store the water influx rates of its neighbors (figure 5). These are assumed to remain constant until altered by respective neighbor cells at "significant" events in their state behavior. In a simple reservoir model, these events are: outflow initiation (when a reservoir fills to capacity), and flow cessation (when the reservoir level recedes below capacity). The simulation cycle then appears as:

Concurrently, each simulator:

1. Until it detects a significant event
   computes local model state transition
   (using current influx rates)
2. When significant event is detected
   sends new water outflux rates to

neighbors(as influx rates)
3. When receives new influx rate from neighbor replaces existing influx rate and returns to phase 1.

Flux rates are formulated in water distribution rules as part of the overall model and must satisfy constraints such as mass conservation. Notice that interprocessor communication is necessary only in phase 2. To the extent that this phase is encountered relatively infrequently, the interprocess communication is significantly reduced.

The resulting cell model is a combined continuous and discrete event hybrid that can be nicely described within the DEV & DESS formalism developed by Praehofer (1991). In general, what we have termed significant events above are associated with so-called state events. The latter represent points in the model's continuous state trajectory where some condition, such as a threshold crossing, becomes true. Such mixed formalism models may be represented by pure DEVS models by eliminating the detail of the state trajectories between state events and retaining only the total times taken between such events. Such abstraction may sometimes be performed by closed form analysis but usually requires extensive exercise simulation of the cell model (see Zeigler, 1984 for further discussion).

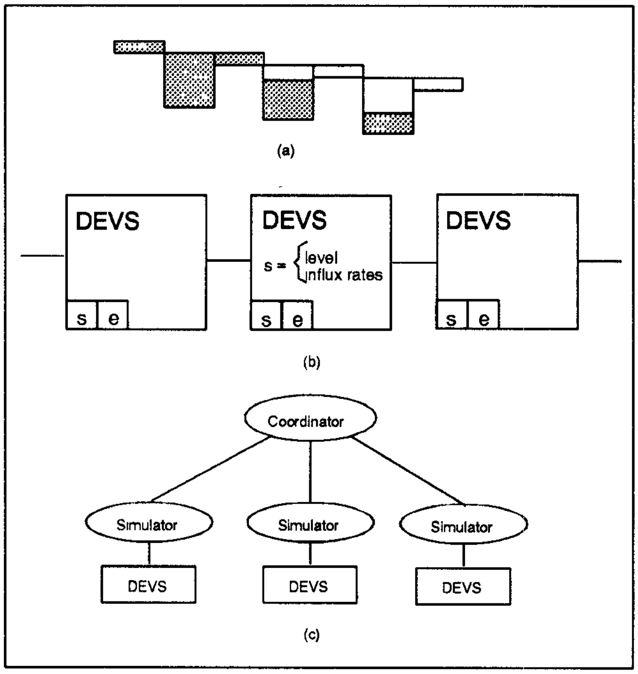

Figure 5: DEVS Cellular Space Representation of Reservoir Cells and Associated Simulation Architectures

5 DISTRIBUTED SIMULATION ALGORITHMS FOR THE DEVS CELLSPACE MODELS

The DEVS cell space formulation makes it possible to consider distributed discrete event approaches to simulation of spatially distributed systems. Typically, processors are assumed to operate asynchronously with message passing intercommunication. In the watershed example, a slow processor (or one executing a more complex cell transition) may sometimes send flux rates that pertain to an earlier time in the behavior of a faster one. One approach is to hold back the faster processors until the slowest has completed its processing so that global time agreement is always maintained. To avoid such drastic slow down, the literature in distributed/parallel simulation categorizes algorithms to speed up execution as "conservative" and "optimistic" (Fujimoto 1991).

We have developed an optimistic approach specifically geared to the DEVS cellspace modelling formalism. Here the objective is not to let the processors get too far ahead of the one with the smallest time of next output (significant event time) and certainly not beyond their own time of next output. In the following the "horizon" is a dynamically computed upper bound on the earliest time of next output:

Concurrently, each simulator:
1. Until it detects a significant event or reaches the horizon, computes local model state transition (using current influx rates).
2. When a significant event is detected, broadcasts local model time as new horizon to all other processors and sleeps.
3. When it receives a new horizon, if this time is less than the current horizon, updates new value and continues (if in Phase 1) to seek next output event.
4. When activated, sends new water outflux rates to neighbors (as influx rates).
5. When it receives new influx rate from neighbor rolls back to appropriate, earlier state if necessary, replaces existing influx rate and returns to phase 1.

The coordinator:
when all processors are sleeping, activates those with minimum time of next output.

This approach can be shown to improve upon the same algorithm with no horizon sharing. Since processor advance is stopped as early as possible, waiting for the slowest processor is minimized as is rolling back to earlier states. Thus the speed-up advantage can be expected to occur where there is significant variation in execution times to next output (see Liao, 1993 for more discussion). More advanced adaptations of disctributed simulation algorithms to DEVS are under study (Praehofer, 1992; B. Lubachevesky, Weiss, and Schwartz 1991; and Wang and Zeigler, 1993).

6 SUMMARY

Working in the context of large scale ecosystems modelling, we have illustrated how DEVS, a general modelling formalism with a system theoretic basis, facilitates combined discrete/continuous modelling of spatially referenced systems. A spatially distributed watershed model demonstrates the utility of linking the DEVS formalism for cellular spaces with geographical information systems to make possible large scale, high resolution modelling and simulation based on voluminous terrestrial data. Approaches to efficient simulation of such DEVS models were discussed based on combining DEVS abstract simulator concepts with distributed simulation methods. Overall, DEVS methodology provides a basis for addressing the critical issues raised by Ho (1989) for supporting the analysis, design and management of discrete event dynamic systems.

7 ACKNOWLEDGEMENT

Our thanks to George Ball, Randy Gimblett and Vincent Copes, professors at the University of Arizona School of Renewable Natural Resources for their help in the ecosystem model formulation.

REFERENCES

Alonso, C. V., and D. G. DeCoursey, "Small watershed model," in *Proceedings of the Natural Resources Modeling Symp.*, pp. 40–46, Pingree Park, CO, 1985.

Ball, G. L., and H. R. Gimblett, "Spatial dynamic emergent hierarchies simulation and assessment system," *Ecological Modelling*, vol. 62, pp. 107–121, 1992.

Band, L. E., and E. F. Wood, "Strategies for large scale hydrologic simulation," *Applied Mathematics & Computation*, vol. 27, no. 1, pp. 23–38, 1988.

Bathurst, J. C. "Physically-based distributed modelling of an upland catchment using the systeme hydrologique europeen," *Journal of Hydrology*, no. 87, pp. 79–102, 1986.

Baumans, Roel M. J., and F. H. Sklar, "A polygon-based spatial (pbs) model for simulating landscape change," *Landscape Ecology*, no. 4, pp. 83–97, 1990.

Baveco, J. M., and R. Lingeman, "An object-oriented tool for individual-oriented simulation: host-parasitoid system application," *Ecological Modelling*, no. 61, pp. 267–286, 1992.

Berry, J. "A mathematical structure of analysing maps," *Environmental Management*, vol. 11, no. 3, pp. 317–325, 1987.

Beven, K. J. "Spatially distributed modeling: conceptual approach to runoff prediction," in *Recent advancesin the modeling of hydrologic systems*, NATO ASI Series, pp. 373–387, Kluwer Academic Pub., 1991.

Costanza, R., F. H. Sklar, and M. L. White, "Modeling coastal landscape dynamics," *BioScience*, vol. 40, no. 2, pp. 91–107, 1990.

Couclelis, H. "Cellular worlds: a framework for modeling micro-macro dynamics," *Environment and Planning*, vol. 17, pp. 585–596, 1985.

Green, D. G. et al., "A generic approach to landscape modelling," in *Proceedings of the Eight Biennial Conference*, pp. 342–347, Canberra, Australia, 1989, Simulation Society of Australia.

Sklar, F. H. et al., "Dynamic spatial simulation modelling of coastal wetland habitat succession," *Ecological Modelling*, vol. 29, pp. 261–281, 1985.

Band, L. E. et al., "Forest ecosystem processes at the watershed scale: basis for distributed simulation," *Ecological Modelling*, no. 56, pp. 171–196, 1991.

Abbot, M. B. et al., "An introduction to the european hydrological system," *Journal of Hydrology*, no. 87, pp. 45–59, 1986.

Running, S. W. et al., "Mapping regional forest evapotranspiration and photosynthesis by coupling satellite data with ecosystem simulation," *Ecology*, vol. 79, no. 4, pp. 1090–1101, 1989.

Farig, L. "A general model of populations in patchy habitats," *Applied Mathematics & Computation*, vol. 27, no. 1, pp. 53–66, 1988.

Freeze, R. A. and R. L. Harlan, "Blueprint for a physically-based, digitally-simulated hydrologic response model," *J. of Hydrology*, vol. 9, pp. 237–258, 1969.

Fujimoto, R. M. "Lookahead in parallel discrete event simulation," in *Proc. of the International Conference on Parallel Processing*, pp. 34–41, Silver Springs, MD, 1988.

Fujimoto, R. M. "Parallel discrete event simulation," *Communications of the ACM*, vol. 33, pp. 30–53, 1990.

Ho, Y. C. "Editor's introduction to special issue on deds," in *Proceedings of the IEEE*, volume 77, pp. 3–6, IEEE Press, 1989.

Hogeweg, P. "Cellular automata as a paradigm for ecological modelling," *Applied Mathematics and Computation*, vol. 27, pp. 81–100, 1988.

Itami, R. M. "Cellular worlds: Models for dynamic conceptions of landscape," *Landscape Architecture*, vol. 78, no. 5, pp. 52–57, 1988.

Liao, C. *Distributed Simulation of Sparse Output DEVS*, Master's thesis, University of Arizona, 1993.

Lubachevesky, B., A. Weiss, and A. Schwartz, "An analysis of rollback-based simulation," *TOMACS*, vol. 1, no. 2, pp. 154–193, 1991.

Morris, E. M. "Forecasting flood flows in grassy and forested basins using a deterministic distributed mathematical model," in *Proceedings of the IAHS Int. Symp. on Hydrological Forecasting*, number 129, pp. 247–255, 1980.

Praehofer, H. *Systems Theoretic Foundations for Combined Discrete Continuous System Simulation*, PhD thesis, University of Linz, Department of Systems Theory, 1991.

Praehofer, H. "Distributed discrete event simulation," Technical Report TR-93, University of Linz, 1993.

Preiss, B. "The yaddes distributed discrete event simulation specification language and execution environment," *Distributed Simulation*, vol. 89, pp. 139–144, 1989.

Rozenblit, J. W., and P. Janknski, "An integrated framework for knowledge-based modeling and simulation of natural systems," *Simulation*, vol. 57, no. 3, pp. 317–336, 1990.

Savatsky, K., and R. G. Reynolds, "A hierarchical model of the evolutionof cooperation in cultural systems," *BioSystems*, vol. 23, pp. 261–279, 1989.

Sklar, F. H., and R. Costanza, "The development of dynamic spatial models for landscape ecology," in *Quantitative methods in landscape ecology*, Springer-Verlag Ecological Studies, pp. 239–288, Springer-Verlag Pub., 1991.

Vasconcelos, M. J. *Modeling Spatial Dynamic Ecological Processes with DEVS-Scheme and Geographic Information Systems*, PhD thesis, University of Arizona, School of Renewable Natural Resources, 1993.

Vasconcelos, M. J., and D. P. Guertin, "Firemap-simulation of fire growth with a geographic information system," *International Journal of Wildland Fire*, vol. 2, pp. 87–96, 1992.

Vasconcelos, M. J., and B. P. Zeigler, "Simulation of forest landscape response to fire disturbances," *Ecological Modelling*, vol. 65, pp. 177–198, 1993.

Wang, Y., and B. P. Zeigler, "Devs based simulation on a simd massivelly parallel architecture," *Journal of Discrete Event Dynamic Systems*, 1993.

Woolhiser, D. A. "Hydrologic and watershed modeling - state-of-the-art," *Trans. Am.Soc. of Ag. Engrs.*, pp. 553–559, 1973.

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

Zeigler, B. P. *Multifacetted Modelling and Discrete Event Simulation*, Academic Press, Orlando, Fl., 1984.

Zeigler, B. P. *Object-Oriented Simulation with Hierarchical, Modular Models*, Academic Press, San Diego, Ca., 1990.