SIMULATION IN CIM AND ARTIFICIAL INTELLIGENCE TECHNIQUES

PROCEEDINGS
OF THE
EUROPEAN SIMULATION MULTICONFERENCE
JULY 7-10, 1987
WIRTSCHAFTSUNIVERSITÄT
VIENNA, AUSTRIA

EDITED BY
J. RETTI
K.E. WICHMANN

ORGANIZED BY ASIM FA 4.5
THE VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS ADMINISTRATION
THE ÖSTERREICHISCHE COMPUTER GESELLSCHAFT
AND THE ÖSTERREICHISCHE GESELLSCHAFT FÜR ARTIFICIAL INTELLIGENCE
IEEE (TC IN SIMULATION)
IN COOPERATION WITH DBSS, ISCS, SIMS, UKSC, JSST.
AND SOGESCI
WITH THE FINANCIAL SUPPORT OF
THE SOCIETY FOR COMPUTER SIMULATION (SCS)

Printed in Belgium

AI TECHNIQUES

STATE OF THE ART PRESENTATION

AI's Role in the Control of Systems 165
Francois Cellier and B. Zeigler

AI & SIMULATION METHODS

Declarative (Non Procedural) Approach in Simulation Languages 175
Albert van der Horst

KNOWLEDGE ACQUISITION FOR EXPERT SYSTEMS

Automatic Updating of the Knowledge Base of an Expert System for the Selection of Optimal Correlations for Physico-Chemical Properties 183
V.G. Dovi, O. Paladino and Bartolomeo Canepa

APPLICATIONS

Intelligent Decision Support for the Management of Hazardous Substances: Symbolic Simulation of Chemical Production Processes 191
L. Winkelbauer and K. Fedra

Complex Models of Tree Development Dynamics, and the Simulation of Tree Response to Pollution Stress 197
Hartmut Bossel, Holger Krieger, Heiner Schaefer and Norbert Trost

Modelling of Open Surface Canal Nets 202
D. Singer and P. Moritz

Fuzzy Rule Based Simulation with Economic Applications 204
Andreas Geyer-Schulz and Alfred Taudes

QUALITATIVE MODELLING

QSOPS: A Workbench Environment for the Qualitative Simulation of Physical Processes 213
Alfred D. Round

MISCELLEANEOUS

A System Teaching a Graphoanalytical Model of Human Operative Memory 221
Simeon Jordanov Mrchev

AI's Role in Control of Systems: Structural and Behavioral Knowledge

Francois E. Cellier and Bernard P. Zeigler
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721

ABSTRACT

This paper surveys the role of Artificial Intelligence techniques in the control of real systems. A broad view of the control problem is taken which includes aspects such as long term planning, system management, fault diagnosis, user interfacing, etc. Within this scope, we identify opportunities for research combining AI and simulation methodologies including: comprehensive frameworks for knowledge representation, knowledge based automated model synthesis and system design, and combining qualitative and conventional systems modelling for multilevel control.

INTRODUCTION

The role of AI applications can be understood by starting with our desire to influence and control nature to suit our purposes. To do this, we try to predict the behavior of real systems, and how our actions influence this behavior. If our predictions are reliable, we can take effective action. Models of real systems help us to predict their behavior and find actions that will achieve our goals. We build models by observing real systems and organizing the results of our observations in a useful form. We test models by comparing their predictions against new observations, and modify them if significant discrepancies are noted. Once validated, models can be used to design new systems and to control the behavior of existing ones. Having knowledge about a system and having a valid model for it are basically two different ways of saying the same thing.

Control of a system can be illustrated by an example. Piloting an ocean liner begins by planning a course to sail to a destination. We employ our knowledge, or model, of ocean currents and winds, and of the ships capabilities, to lay out the course that will make our journey pleasant and get us to where we want to be on time. If our knowledge is faulty, i.e., our model's predictions are not borne out, our voyage will not be as smooth as planned. Normally, the winds and currents are close to what we expected, and all that is required is to steer the ship in the right direction despite small disturbances that are ever present. If something major goes wrong, and we find ourselves off course, we try to find out what it is, and having done so, take corrective action.

We see that control of nature, broadly interpreted, involves system design, long range planning, and short range monitoring and compensation. It also involves diagnosis (finding out what went wrong) and repair. Models are essential to effective handling of each aspect, but they are different. To steer the ship, we need to know how it will respond to movements of the steering wheel and changes of motor speed. Classical automatic control systems employ dynamic models for this purpose. However, such models are not suitable for the other aspects of system control. It is in the latter aspects, that the knowledge representation schemes and concepts of AI come primarily into play.

THE CONTROL OF COMPLEX SYSTEMS

Different aspects of the (local and global) control of complex technical processes, the types of knowledge used for these control purposes, and the tools that are primarily employed in these aspects are summarized in *Table 1*.

It can be seen that a distinction is made between structural and behavioral knowledge. Roughly, structural knowledge refers to knowing how a system is constructed. The behavior of a system refers to what it does rather than how it is constructed. Behavioral knowledge refers to knowledge of its behavior deduced from observations that are made on the system over time.

It is in the use of structural knowledge that AI has made its largest impact. For example, in long range planning, such as for the ocean liner's route, we must have available an accurate map of the world showing the relatively fixed patterns of currents and winds. Such structural knowledge can be contrasted with the on-line measurements of current and wind velocities that are fed to the automatic steering control system (behavioral knowledge). Similarly, to diagnose a problem in the ship's motor, we must know the details of the motor's construction, what parts are most likely to fail (structural knowledge), and how these faults manifest themselves as observable symptoms (behavioral knowledge).

Table I. Aspects involved in the control of complex multivariable systems, the types of knowledge that are appropriate, and the associated tools that make use of this knowledge.

Aspect of control	**Type of knowledge**	**Tool or instrument**
system design	structural/behavioral	expert system/simulation
prediction	structural/behavioral	expert system/simulation
long range planning	structural/behavioral	AI planner/simulation
state monitoring	behavioral	sensors
state reconstruction	behavioral	observers/filters/estimators
control of state variables	behavioral	feedback control system
(e.g. position and velocity control)		
fault diagnosis (find out what went wrong)	structural/behavioral	expert system/simulation
fault correction (system repair)	structural	expert system
advise human operator	structural/behavioral	consultation system

While AI has brought us the ability to employ structural knowledge, this is not to say that behavioral knowledge is unimportant. Neither would it be correct to assume that AI has no rôle to play in the processing of behavioral knowledge, or that structural knowledge processing and behavioral knowledge processing are two separate issues that are completely decoupled from each other. An effective global control system must be able to deal with both forms of knowledge simultaneously, as is evident from *Table I*.

While structural knowledge concerning a process is normally time-invariant, and thus static, behavioral knowledge is always time-dependent, and thus dynamic. However, even the structural properties of a control system may experience changes, in particular as a result of the occurrence of a fault.

Let us assume, our ocean liner was involved in a collision with an iceberg. After the collision has taken place, we can no longer trust that our static structural information about the system is still completely correct, and we require a means to qualitatively verify certain hypotheses about the ship's way of functioning on the basis of new structural as well as behavioral observations.

MODEL AND KNOWLEDGE REPRESENTATION

The models that are being employed by simulation programs (e.g. for the purpose of the design of feedback controllers) and by expert systems (e.g. for the purpose of city-wide traffic control) are traditionally very different and usually incompatible in many dimensions such as formalism, level of aggregation, etc.

Simulation models for the purpose of control system design are usually specified either as state-space models or by means of transfer functions. If the system variables are viewed as varying continuously over time, the models are expressed as a set of first-order differential equations in the time domain or by means of one or several transfer functions in the frequency domain (Laplace transformation). If the system variables are viewed as changing at discrete time instants only (time slicing), the models are written as a set of first-order difference equations in the time domain or by means of one or several transfer functions in the frequency domain (z-transform). Sequencing control problems (e.g. for flexible manufacturing systems) are more often represented by discrete event models. Although, few algorithms for control system design exist for these types of models, there is increasing interest in developing a suitable control theory for this domain (e.g. a special issue of IEEE Proceedings will be devoted to the Dynamics of Discrete Event Systems).

In contrast, models employed by rule-based controllers are commonly static in nature. The knowledge processed by these control systems is mostly structural. If behavioral knowledge is utilized at all, it occurs in the form of statistical data. If time does appear as a variable, it is used only to switch between several (in themselves static) models. A typical example of such a control system might be the traffic control of an urban area, where the structural knowledge is represented by a street map, and the behavioral knowledge is represented by an origin-destination matrix (OD-matrix) which contains statistical data concerning the frequency of particular traveling needs by the traffic participants. Time appears in the form of different OD-matrices being used at different times of the day, and at different days of the week (behavioral knowledge), and in the form of incorporation of ongoing street constructions (structural knowledge). The expert system allows the construction engineer to judge the effects on the statistical distribution of individual traffic loads, of a scheduled construction activity and/or plans for new streets to be added to the street plan.

We suggest that the current state-of-the-art in both areas has not reached a level of maturity yet. Although the way how knowledge is utilized by the two types of knowledge processing systems (simulation system and expert system) may differ drastically, they should still be able to refer to and extract data from one and the same knowledge base. They should also be able to communicate with each other through this knowledge base by modifying data entries in it.

Local control (micro-control) of individual subprocesses can be improved by providing global information about the overall system performance. Currently, adaptive controllers identify the parameters of a system in order to adapt the control parameters to varying environments. In like manner, the structural knowledge about the overall system can be used to switch between entirely different control strategies. Conversely, global control (macro-control) of the overall process can be improved by making available to it time-varying properties of the local subsystems. For example, overall traffic control can be improved by providing the controller with information about momentary congestions in the system (besides broadcasting this information to the traffic participants).

To implement this approach, we must be able to represent knowledge about the system so that it can be extracted by different processes for different purposes. This requirement has been a major obstacle in the past. Expert systems require structural knowledge of the system to be stored in the form of abstract data items. So-called knowledge based simulation models[1,2] are developed on the basis of such a structure representation. Unfortunately, execution of simulation runs using these models tends to be slow since the simulation engine must be able to manage both structural and behavioral changes as potentially present. Especially, knowledge based continuous-time models are painfully slow to execute as the data structures representing the differential equations are being re-interpreted during each integration step.

Run time improvement can be obtained by extracting only the structural knowledge required for the task at hand from a more comprehensive representation. Techniques have been proposed, called model pruning[3], that are capable of automatically performing such extraction. After the model has been pruned, the resulting (simplified) model is still stored in an abstract data format. At this stage, an additional preprocessing step can be used to synthesize a coded model from components in a model base (DEVS-Scheme[4,5]). Other possibilites are to generate the model in the form of a SIMSCRIPT II.5 (discrete event) or ACSL (continuous-time) program that can be compiled and linked. Alternatively, a *direct executing* SLAM (discrete event) or DESIRE (continuous-time) program can be generated that can be immediately run. DESIRE[6] is a continuous system simulation language able to process state-space models consisting of mixed differential equations and difference equations. DESIRE has been designed with the goal of maximizing the program execution speed under the constraint that no (time-consuming) compilation and linkage is needed. An ultra-fast microcompiler translates the differential equations into an internal data structure while the surrounding flow control statements are executed in an interpretive manner. For models consisting of up to 100 differential equations, the compilation time needed is below 0.8 sec. on an IBM PC/AT.

It should be remarked that the manner in which simulation programs were traditionally specified, e.g. by means of a state-space model of the process to be controlled, is not the only possible means to represent the model structure for simulation processing. Indeed, this may not even be the most adequate way of representing the model structure for some of the «simulation tasks» identified in *Table I*, e.g. for the purpose of determining global behavioral patterns. In such a case, it may be advantageous to process the behavioral information in a *qualitative* rather than a *quantitative* manner. A methodology for extracting qualitative information from behavioral system data, and a means to construct qualitative (inductive) input/output models from behavioral information is currently being investigated[7,8].

MULTIPLE USES OF THE KNOWLEDGE BASE

Yet another contribution of the emerging AI technology deserves to be mentioned. It concerns additional uses of the same knowledge base besides that of control.

In retrospect, it seems like a very natural thing to do: separate the knowledge initially given to a program, and acquired during its operation, from the procedures for processing it. Yet the concept that knowledge is somehow different from the rest of computer code and could be treated as a thing in, and of, itself, was slow in coming – its power is only now becoming widely appreciated. When knowledge is interlaced with the rest of the code of a program, there is no easy way of using it for more than one related purpose. For example, when a conventionally programmed logic controller encounters an exceptional condition in the process under control, it quickly aborts all ongoing activity. To restart the process, a human operator must be able to figure out what went wrong.

Little help is afforded by the controller, since it cannot report the information it has acquired about the current state of the process in the course of its operation. However, this information is just what the operator needs to know to be able to set the process back on course.

In contrast, a controller designed with AI concepts has a separate component, called a knowledge base, which stores both structural and behavioral knowledge. This knowledge is then usable not only for the primary purpose of direct control but also for such purposes as providing the operator with a good picture of the current state of the process as well as helping to diagnose system failures as they occur.

Indeed, any or all of the aspects shown in *Table 1* may be implemented by different procedures, all of which share the same knowledge base. In this way, a system can be designed which is not only more effective in carrying out its primary functions, but also provides much greater help to the user than was possible before.

COMBINING AI AND SIMULATION METHODOLOGY

There is much interest in combining Artificial Intelligence (AI) and simulation methodologies. AI knowledge representation schemes go beyond classical model formalisms in allowing dimensions of representation such as inference of new knowledge, associative and other access to existing knowledge, matching of patterns, and meta-knowledge[9,10,11]. Although such schemes can organize much of the knowledge about systems that was unrepresentable with dynamic formalisms, they are not adept at representing the dynamics that the latter was invented for. Thus, there needs to be a paradigm whose scope of representation includes both classical and AI schemes. This is an area of research with at least two sources of inspiration. One stems from the hypothesis that for a computer to reason about physical systems, it must have more of a qualitative and common sense representation than exists in the classical modelling formalism[12,13]. However, in attempting to capture common sense knowledge about physical systems, qualitative modelling may drastically coarsen the state descriptor space from the real numbers to small discrete sets. Accordingly, much ambiguity arises in generating the behavior (reasoning) of such models. Recent approaches use symbolic means to summarize dynamic system behavior rather than to derive it[14,15].

Another source of inspiration for a wider paradigm comes from computer simulation. Here the world views of discrete event simulation have been found to be highly compatible with the representation schemes of Artificial Intelligence[16]. Object-oriented programming[17,18] can be viewed as providing a computational basis for knowledge representation by allowing the programmer to associate methods with objects organized in taxonomic classes. Such methods can perform operations on the global object state (the ensemble of its slots), and invoke each other by passing messages. Actually already in 1967, the discrete event simulation language SIMULA introduced class inheritance and association of both procedures and data structures with class instances. It is not surprising therefore, that languages are being developed to express both the dynamic knowledge of discrete event formalisms and the declarative knowledge of AI frame paradigms[19]. Other articles published in the same volume are also relevant in this context.

Among the many opportunities for research are the following:

- knowledge representation schemes encompassing both classical dynamical formalisms and AI representation schemes,

- use of knowledge representation to organize model bases and direct automated model synthesis, and

- approaches to combine qualitative and simulation models in system control.

We shall briefly describe our own research in these areas.

MIXED KNOWLEDGE REPRESENTATION SCHEMES

A knowledge representation scheme to combine the expressive power of both classical dynamical system formalisms and the newer AI based schemes has been proposed[3,20,21]. The system entity structure incorporates decomposition, taxonomic, and coupling knowledge concerning a domain of real systems. An extensive entity structure was constructed for the domain of local area networks (LANs) and their interconnection via gateways[22,23,24].

ORGANIZING MODEL BASES AND AUTOMATED MODEL SYNTHESIS

An intelligent manager for model specification, storage, and retrieval has been developed to mediate between the user and the DEVS-Scheme simulation environment[4,5]. The manager contains a system entity structure manipulation module, called ESP (Entity Structurer and Pruner). In such a system, one uses ESP to prune a comprehensive entity structure according to the objectives of the modelling study. This results in a so-called pruned entity structure that specifies a hierarchical discrete event model. Upon invoking the «transform» procedure, the system searches the model base for components specified in the pruned entity structure. When all components are found, the system synthesizes the desired model by coupling them together in a hierarchical manner employing coupling knowledge contained in the pruned entity structure. The result is a simulation model expressed in DEVS–Scheme which is ready to be executed to perform simulation studies.

Current research seeks to employ the system entity structure concepts to develop a model-based system design methodology[25,26]. In such a methodology, the designer would be aided by an intelligent advisor in pruning the entity structure in order to meet his design objectives, taken individually, and in trade-off combinations. Structural constraints are associated with the decompositions in the entity structure and give rise to production rules for automated synthesis of models retrieved from the design model base. Synthesized models are amenable to simulation experimentation in experimental frames associated with stated design objectives.

COMBINING QUALITATIVE AND SIMULATION MODELS IN CONTROL

Just as classical dynamic models are employed in automatic feedback control, so can qualitative models be employed at levels of supervision above the conventional levels. Such qualititative models may be developed to support rule-based detection of global patterns in controlled system structure changes[8]. For example, such production rule systems (expert systems) may monitor sensor outputs (usually in highly aggregated form) for trends that indicate that the system trajectory is heading toward a disaster state. Such models may be built using on line data with automated inductive modelling techniques[7,8], or adaptive system approaches[27].

Simulation models can be combined with such rule-based predictors in the following ways:

- the simulation model, assumed to be valid, is employed to produce behavior for off-line identification of the qualitative model, and hence of the rule-based predictor[2],

- the state of the simulation model is synchronized with the controlled system and runs in faster than real time; the output is fed continuously to the rule-based predictor, thus enabling earlier warning of impending critical states and use of higher level advisor systems for stragegy selection[28].

QUALITATIVE SIMULATION FOR GLOBAL PATTERN EXTRACTION

In this paragraph, it is our aim to demonstrate the potential power of qualitative simulation for the purpose of global pattern identification in continuous-time systems.

For this purpose, let us consider the simple state space model:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -4 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

We simulated this system *quantitatively* over time using a random number generator for the input signal *u*, and recorded both the input signal and the three state variables at predetermined instants of time. The simulation results were collected into a matrix of dimensions 101x4, where each row denotes one recording, and each column denotes one variable. These simulation results were to represent our «measurement data». Thereafter, the variables were quantized (recoded) into a finite state space. (We could also say that the data were *categorized*, that is: classified into several categories such as «low», «average», and «high».) The recoded data represent the *qualitative behavior information* about the system to be analyzed.

We then used the first 90 recordings (rows) to automatically generate a *qualitative, inductive input/output model* with optimized forecasting power. For that purpose, we used the general system problem solving (GSPS) formalism[7,8,29,30]. Using this model, we then forecasted (that is: qualitatively simulated the model) over 11 steps which we were able to compare to our «measurement data» as obtained from the previous quantitative simulation, and thereby, we were able to validate our qualitative simulation model.

It turned out that the prediction using this *qualitative model* with some random streams did not contain a single error, while other input streams led to very few differences between the «measured» (that is: quantitatively simulated) and the «predicted» (that is: qualitatively simulated) data streams.

These results encourage us to apply the same methodology to more interesting problems. Currently, one of our students works on the quantitative simulation of an aircraft where «accidents» are being simulated. It will be the task of our global controller to:

(1) identify «on the fly» *that* something went wrong,

(2) determine a new qualitative model of the modified aircraft (which e.g. may just have lost an engine),

(3) determine which controls are still operational,

(4) identify the effect of these controls on the qualitative model of the damaged aircraft,

(5) compute an optimal control strategy using the forecasting model to e.g. successfully land the damaged aircraft, and finally

(6) apply this optimal control strategy to the «true» aircraft (that is: the quantitative model) to verify the success of the approach.

CONCLUSIONS

This survey has attempted to outline the contributions that artificial intelligence can make to the control of dynamic systems. We have viewed system control within a broad framework so as to put into perspective the areas in which AI and classical control paradigms can be fruitfully combined. Among the many opportunities for research, we have focussed on: development of knowledge representation schemes encompassing both classical dynamical formalisms and AI representation schemes, use of knowledge representation to organize model bases and direct automated model synthesis, and approaches to combine qualitative and simulation models in multilevel system control.

REFERENCES

[1] **Davis, P. K.**, «Applying Artificial Intelligence Techniques to Strategic-Level Gaming and Simulation», In: *Modelling and Simulation Methodology in the Artificial Intelligence Era*, (M. S. Elzas, T. I. Ören, B. P. Zeigler, eds.), North Holland Publishing Corp., Amsterdam, 1986, pp. 315–338.

[2] **Reddy, Y. V. *et alia*,** «Automating the Analysis of Simulations in KBS», Proc. SCS Multiconference, San Diego, CA, 1985.

[3] **Zeigler, B. P.**, *Multifacetted Modelling and Discrete Event Simulation*, Academic Press, 1984.

[4] **Zeigler, B. P.**, «DEVS-Scheme: A Lisp-Based Environment for Hierarchical, Modular Discrete Event Models», Technical Report: **AIS–2**, CERL Lab., Dept. of Electr. and Comp. Engr., University of Arizona, Tucson AZ 85721, 1986.

[5] **Zeigler, B. P.**, «Hierarchical, Modular Discrete Event Modelling in an Object Oriented Environment», *Simulation*, accepted for publication, 1987.

[6] **Korn, G. A.**, «Control System Simulation on Small Personal Computer Workstations», *Inter. J. Modelling and Simulation*, accepted for publication, 1987.

[7] **Cellier, F. E.**, «Qualitative Simulation of Technical Systems Using the General System Problem Solving Framework», *Inter. J. General Systems*, accepted for publication, 1987.

[8] **Cellier, F. E., and D. W. Yandell,** «SAPS-II: A New Implementation of the Systems Approach Problem Solver», *Inter. J. General Systems*, accepted for publication, 1987.

[9] **Kerckhoffs, E. J. H. and G. C. Vansteenkiste,** «The Impact of Advanced Information Processing in Simulation – An Illustrative Review», *Simulation*, **46**(1), 1986, pp. 17–26.

[10] **Robertson, P.**, «A Rule Based Expert Simulation Environment», In: *Intelligent Simulation Environments*, (P. A. Luker and H. H. Adelsberger, eds.), Simulation Series, **17**, SCS, San Diego CA, 1986.

[11] **Futo, I.**, «Combined Discrete/Continuous Modeling and Problem Solving», In: *AI, Graphics and Simulation*, (G. Birtwistle, ed.), SCS, San Diego CA, 1985.

[12] **Bobrow, D. G.**, *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge MA, 1985.

[13] **Rajagopalan, R.**, «The Role of Qualitative Reasoning in Simulation», In: *Artificial Intelligence in Simulation*, (G. C. Vansteenkiste, E. J. H. Kerckhoffs and B. P. Zeigler, eds.), SCS Publications, San Diego CA, 1986.

[14] **Hardt, S. H.**, «Aspects of Qualitative Reasoning and Simulation for Knowledge Intensive Problem Solving», In: *Modelling and Simulation Methodology: Knowledge Systems*, (M. S. Elzas, T. I. Ören, B. P. Zeigler, eds.), North Holland Publishing Corp., Amsterdam, to appear, 1988.

[15] **Reddy, Y. V., *et alia*,** «The Knowledge-Based Simulation System», *IEEE Software*, March 1986, pp. 26–37.

[16] **O'Keefe R.**, «Simulation and Expert Systems – A Taxonomy and Some Examples, *Simulation*, **46**(1), 1986, pp. 10–16.

[17] **Adelsberger, H. H. *et alia*,** «Rule Based Object Oriented Simulation Systems», In: *Intelligent Simulation Environments*, (P. A. Luker and H. H. Adelsberger, eds.), Simulation Series, **17**, SCS, San Diego CA, 1986.

[18] **Bobrow, D. G. and M. J. Stefik,** *The LOOPS Manual*, Xerox Corporation, Palo Alto CA, 1983.

[19] **Klahr, P.**, «Expressibility in ROSS, An Object-Oriented Simulation System», In: *Artificial Intelligence in Simulation*, (G. C. Vansteenkiste, E. J. H. Kerckhoffs and B. P. Zeigler, eds.), SCS Publications, San Diego CA, 1986.

[20] **Zeigler, B. P.**, «Knowledge Representation from Newton to Minsky and Beyond», *Applied Artificial Intelligence*, **1**(1), 1987, pp. 87–107.

[21] **Zeigler, B. P. and G. Zhang,** «Formalization of the System Entity Structure Knowledge Representation Scheme: Proofs of Correctness of Transformations», In: *AI, Simulation, and Modelling*, (L. Widman and D. Reidel, eds.), in preparation, 1987.

[22] **Zeigler, B. P. and S. Sevinc,** «Concepts for Design and Simulation of Local Area Networks», Technical Report: **AIS–1**, CERL Lab., Dept. of Electr. and Comp. Engr., University of Arizona, Tucson AZ 85721, 1986.

[23] **Rozenblit, J. W. et alia,** «Knowledge–Based Design of LANs Using System Entity Structure Concepts», Proc. Winter Simulation Conf., 1986.

[24] **Sevinc, S. and B. P. Zeigler,** «Entity Structure Based Design Methodology: A LAN Protocol Example», Technical Report: **AIS–4**, CERL Lab., Dept. of Electr. and Comp. Engr., University of Arizona, Tucson AZ 85721, submitted to: *IEEE Trans. Software Engineering*, 1987.

[25] **Rozenblit, J. W. and B. P. Zeigler,** «Design and Modelling Concepts», In: *Encyclopedia of Robotics*, (R. Dorf and S. Nef, eds.), J. Wiley & Sons, to appear, 1987.

[26] **Rozenblit, J. W. and B. P. Zeigler,** «Multifacetted Methodology: A Framework for Knowledge–Based Simulation Modelling», Proc. Eastern Simulation Conference, Washington, D. C., 1987.

[27] **Stackhouse, C. and B. P. Zeigler,** «Learning Plateaus in Adaptive Rule Based Systems», In: *Modelling and Simulation Methodology: Knowledge Systems*, (M. S. Elzas, T. I. Ören and B. P. Zeigler, eds.), North Holland Publishing Corp., Amsterdam, to appear, 1988.

[28] **Lemon, H.,** «Comax: An Expert System for Cotton Crop Management», *Science*, **233**, 1986, pp. 29–33.

[29] **Klir, G. J.,** *Architecture of Systems Problem Solving*, Plenum Press, New York, 1986.

[30] **Cellier, F. E.,** «Prisoner's Dilemma Revisited: A New Strategy Based on the General System Problem Solving Framework», *Inter. J. General Systems*, accepted for publication, 1987.