Object based dynamic modeling

Lars Vavik
Stord/Haugesund College
N-5414 Rommetveit, Norway
Phone: +47 5349 1379
Fax: + 47 5341 0160
Internet: larsv@stordlh.no

Magne Myrtveit
ModellData AS
N-5120 Manger, Norway
Phone: +47 5637 4009
Fax: +47 5637 3500
Internet: magmy@modeld.no

(ABSTRACT)

This research is supported by the Norwegian Research Council, Grant no. 31867/223-1994

This paper reports on collaborative action research partnerships with corporations to develop new tools to accelerate learning, and test these tools in real organizations.

In the collaborative program "From Systems Thinking to Praxis" several corporations are facing the challenge of building a virtual model to mirror the companies' business strategies. These models have to meet a level of functionality which allow managers to compare the output of the model with real data from the organization. On the other hand, the models have to be flexible enough to give managers a laboratory for business process redesign.

The focus in this paper is on the laboratory as an exploratory learning environment. The work is based both on modeling as a way of learning, and learning from running simulations (management flight simulators). The challenge is to build a bridge between the functionality and the transparency of the model. A combination of the flexibility of the modeling toolkit and the constraints of a simulation applications is needed.

The computer based learning environment presented here is based on a state of the art toolkit for constructing system dynamic models (graphical flow diagram, arrays, OLE links, DDE data transfers). To meet the new requirement, major extensions were made to the toolkit in order to allow for concrete objects in addition to the general and abstract objects of accumulator–flow diagrams. The resulting system opens up for modeling in terms of objects within the problem area (for example banks, markets, capital, transportation), and at the same time gives the possibility to use the basic blocks (accumulators and flows).

A management flight simulator based on high level objects renders the possibilities to change the model structures, not only parameters.

The value of a modeling process based on *object based dynamic modeling* is measured as the probability that a manager will grasp the result from a single analysis and act on it. With the object based dynamic modeling approach we challenge past conclusions that there is a tradeoff between model complexity and the value to the user – a compromise between functionality and transparency.

Keywords: system dynamics, objects, computer aided modeling, exploratory learning

Limits to learning in computer based environments

When we use the term "learning environment" in connection with computer-based simulation, we refer to a computer based arena where people are free to learn, using their own style at their own pace, and in a sequence that they determine themselves. A simulation application and a modeling toolkit provides the "engine" for learning within this type of environment. In research on instructional science, simulation and modeling have been distinguished as two separate learning environments (Elsom-Cook, 1990; Sellman, 1992):

Simulations are based on a mathematical model of some system. Unlike modeling systems, the model is not explicitly available to the learner and the model structure cannot be changed. The learner has some sort of representation of the behavior of the model, and can manipulate certain variables and observe the effect on other variables.

Modeling systems allow the learner to construct mathematical models of a system and manipulate those models to see how they respond under various circumstances.

Limits to learning in the simulations application

Learning environments utilizing simulation applications have found a stable niche as an instructional measure in varied areas. There is substantial interest in the use of computer-based simulation in industrial training. By reviewing proceedings from system dynamics literature there seems to be a growing interest in transferring models from modeling systems into specific simulation based applications. In the system dynamics tradition such applications are called "models with games interfaces" or "management flight simulators" (Andersen et al. 1990; Bean, Diehl & Kreutzer 1992; Kreutzer et al. 1992; Morecroft 1992; Davidsen 1994; Senge 1994). The architecture of a management flight simulator follows the same structure as for traditional computer based instruction with simulation. Learning based on these systems fits well into a general pool of problems.

The details of the system complexity are hidden from the user by an apparently "simple" user interface. There are limitations to the usefulness of simulators as stand-alone tools. The user is only a consumer of the model on which the simulator is based, and does not participate in the model creation process. The lack of involvement in conceptualizing the model may result in a shallow understanding of the dynamics of trial-and-error experimentation (Kim 1990). A deeper understanding of the underlying structure requires more than playing the game repeatedly. There is need for more explicit discussion regarding the theory underlying the operational interface. We fear that the learner presses buttons and moves the slide bars as if he were simply playing a computer game. Little substantial learning occurs in this situation. Users may learn how to win a game, but they probably don't have a clue of what is really going on below the surface (Bakken 1989; Senge 1994).

The implication is that we can no longer afford to design the model and simply track on an interface afterwards. This agrees with Brown's analysis (1989) of the requirement for producing a "glass-box technology". He argues that systems need to be made transparent in three senses: *domain transparency*, allowing the user to see "through" the tool; *internal transparency*, allowing the user to see "into the tool"; and *embedding transparency*, allowing the user to see the relationship of the technology to the lager context of the interaction.

There are several attempts to solve these problems by embedding the simulation in an Intelligent Tutoring System (ITS). Some good examples of Intelligent Simulation Learning Environments (ISLE) are reported for operational training (SHERLOCK, Lajoei & Lesgold 1989, 1991).

As a stand alone tool, this means that we should either build an instructional shell around the application, or the systems should be implemented in an organization with careful advice from a consultant. Simulation applications in organizational learning have been used more as icebreakers, just to pin point some generic structure or simply to introduce a general dynamic problem. In business process redesign there is a need to change the structure of the model, not only the parameters. This requirement can hardly be supported from an ISLE shell.

Limits to learning in the modeling environment

A modeling system gives possibilities to learn about a domain by participating in constructing the model. This is a constructivists vision of the learner as a social tool user. The computer environment itself does not take control of the interaction, and does not react in the "intelligent" way (reasoning about its own behavior) we expect of a simulation application embedded in an ITS architecture. Some researchers (S. Papert 1980; John Holt, Tavistock Institute) suggest that the combination of student and environment is all that is required.

Certainly, the computer modeling process can play an important role in facilitating a group of managers "to make up their own minds" about a system. Lane (1994) describes a shift in modeling style away from traditional expert consultant toward modeling as learning. Modeling becomes an integral part of the management discussion. Consultants provide tools that capture and express the mental models of clients. Models are owned, run and interpreted by the clients.

Much effort has been put into the modern modeling toolkits to make the modeling process as understandable and powerful as possible. Significant progress has been made from the early FORTRAN compiler to a modern system dynamics toolkit. Modern modeling environments offer several valuable instructional measures and concepts. Examples include the syntax of the modeling language; the use of a conceptual accumulator–flow framework to present the structure of models; the integration of diagram and equations to make a graphical modeling environment.

But even the accumulator–flow framework for model conceptualization is rigorous and precise. It is also abstract, and in many cases not the "natural" way for the less-proficient modeler to think about system structure (S. Peterson, 1994, p. 299). The implication is that when it comes to detailed algebraic modeling with the basic building blocks, the modeling process is often taken over by technical experts. Morecroft et al. (1994) stated that "... a small team took part in this phase. It is difficult to imagine how one would have engaged the full team in this more detailed work".

Several research papers referring to Papert's micro worlds (1980) and the constructivistic tradition as a theoretical fundament for "modeling as learning" (Senge, Lane, Morecroft). Unfortunately, many case studies show limited success with the early attempts to design micro worlds. The answer to this might be found in the discussion about "cognitive hooks". The gap between the environment and the existing knowledge is often to great. If we wish to provide environments that are suitably supportive to all learners, we'll have to build environments with "cognitive hooks"; that is, pieces of the environment that a student can hold on to in order to develop better models in a meaningful way.

It is impressive to observe trainees in front of a flight simulator. Even more impressive is the transfer from simulator training to take-off with 250 passengers – without help of an instructor. Observing a team of managers piping accumulator–flow diagrams together in an attempt to build their own management flight simulator, gives mixed impressions.

Redesigning CBLES to Enhance Learning

This research project aims at building a bridge between *qualitative mapping* of organizational processes and *quantitative modeling* of systems. A collaborative action research partnership is set up with a group of senior managers and planners from a major oil company. The aim is to build virtual worlds to mirror and develop business strategies.

The requirement

A major challenge is to design a system that provides a library of domain specific building blocks embedded in a modeling tool. The components have to be flexible to give planners and business unit managers the power to build models for describing and redesigning business processes.

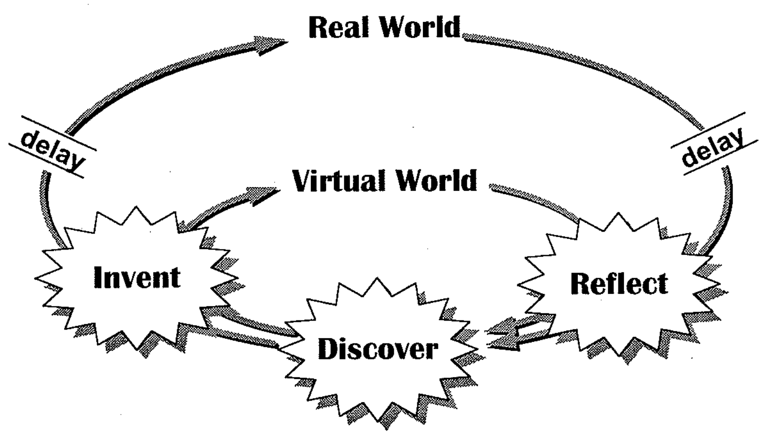

Figure 1: Variant of Argyris and Schon's learning cycle

On the other hand, the models must not just be black-box predictors. It should be possible for the user to compare the output of the model to real business data from the organization (Myrtveit 1995).

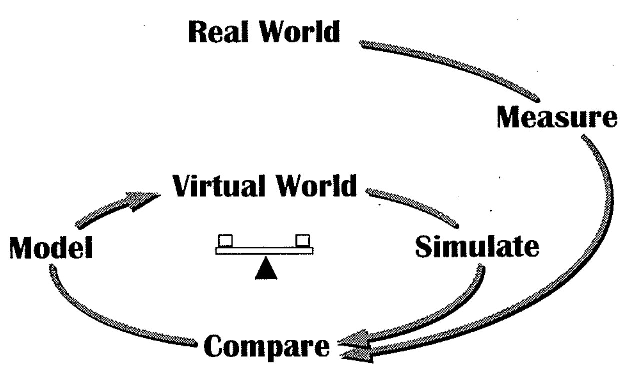

Figure 2: Comparing model output to real business data

The resulting application should be an integral part of the modeling toolkit, allowing users to control, manipulate, and monitor the simulation at different levels of abstraction.

The modeling process and the simulation process correspond to designing organizations and running organizations. Business process redesign is a tool for planning changes to organizations, while total quality management deals with optimizing the way a process is run. Designing new business processes is especially important in a world with rapid changes in technology (Foster 1986). This is where quantum leaps may be done through innovation. The potential of improving processes operation is limited by process constraints (Senge 1990, Limits to success archetype). These relationships are summarized in Figure 3.

Modeling	**Simulation**
Changing structure	Changing parameters
Planning a business	Operating a business
Business process redesign	Total quality management
"Doing the right things"	"Doing things right"
Quantum leaps	Limits to success

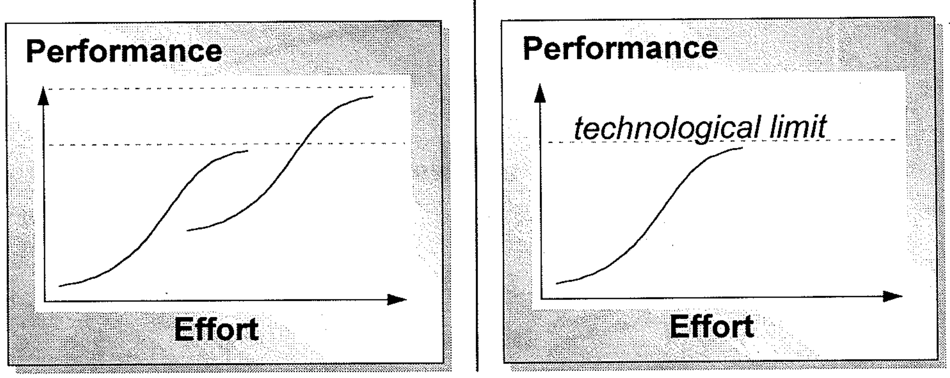

Figure 3: Modeling and simulation; planning and operation

Objects as building blocks

Object orientation is becoming increasingly popular both as a design methodology and as a technology for creating computer software. The use of objects is an elegant way to break the "world" into smaller parts that are easier to handle.

Classification is an important characteristic of objects. This means that objects with the same properties (or attributes) and operations are grouped into a class. An attribute or operation that is local to a class, can be hidden (encapsulated) inside the class.

Only in rare circumstances will an accumulator–flow diagram represent a natural object mapping of a system. (A diagram of physical flows through pipes is an exception.) In general, accumulator–flow diagrams focus on object attributes and relationships between attributes. This is only natural, as the main reason for using accumulator–flow diagrams is to describe dynamic relationships between attributes of a system and deduce the resulting behavior over time. Providing higher level objects may be a way to make modeling useful to non-modelers.

Choice of objects

The topology of a system is in this context defined as the physical arrangement of the building blocks which describe the total system. In general, objects depend both on the system to be modeled and the use of the model.

It is important that objects are chosen from the problem domain – and not from the technology (implementation language, software tool). In this way model users can relate to the models without knowledge about the underlying technology (for example, accumulator–flow diagrams). Figure 4 illustrates how a palette with high-level objects (building blocks) can be combined to describe a problem within a given area (for example manufacturing).

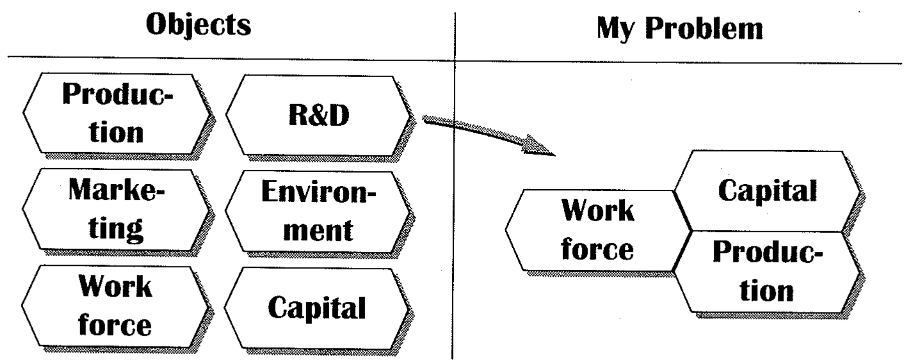

Figure 4: Describing a problem by combining high level objects

The interface to objects

It is common that a model can be presented through different views, for example an equations view (textual equations), a diagram view (accumulator–flow diagram), and an application interface (micro world). For object based systems one interface is used to *define* objects, and another to *display* objects. The use of so-called *property sheets* has become the standard way of defining objects in a graphical user interface like Windows and OS/2. A property sheet is a definition dialog box for an object, and contains controls for determining the attributes of an object. When opening the property sheet for a given object, only attributes for that object will be made available for the user. This ensures that property sheets of high-level objects will not get cluttered with details from lower levels.

Example – A micro world for the gas industry

The objective of this project was to study alternative ways of developing the gas fields of the North Sea. The complex dynamics of markets, transport and processing led to a simulation approach. A decision was made to create a micro world, where planners could build alternative infrastructures and simulate the consequences. A major requirement was that it should be easy to compare alternative strategies. Therefore all reports were designed to display values from two simulation runs. Consistent use of color was used

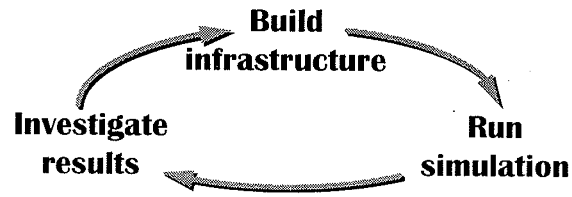

Figure 5: Using the gas micro world

to identify current run and reference run throughout the system (red: current run, gray: reference run). A typical use of the gas micro world involves three steps in a closed loop (Figure 5).

Constructing a model in the micro world

The user is modeling the micro world using high level objects from the problem domain. Objects include: reservoir, well, pipe, processing facility, compressor, joint, storage, and market.

Each object is associated with an icon that is easily recognized by the typical user. Objects are made available for the user as building blocks in a toolbar (see left hand side of Figure 6). Objects are moved into the map according to standard Windows conventions (point, click, drag).

The user opens the property sheet of an object by double-clicking on the object. Each property sheet lists parameters of an object. Parameters are chosen to give enough flexibility for the user to investigate the main issues in the problem area. At the same time care has been taken to avoid unnecessary detail.

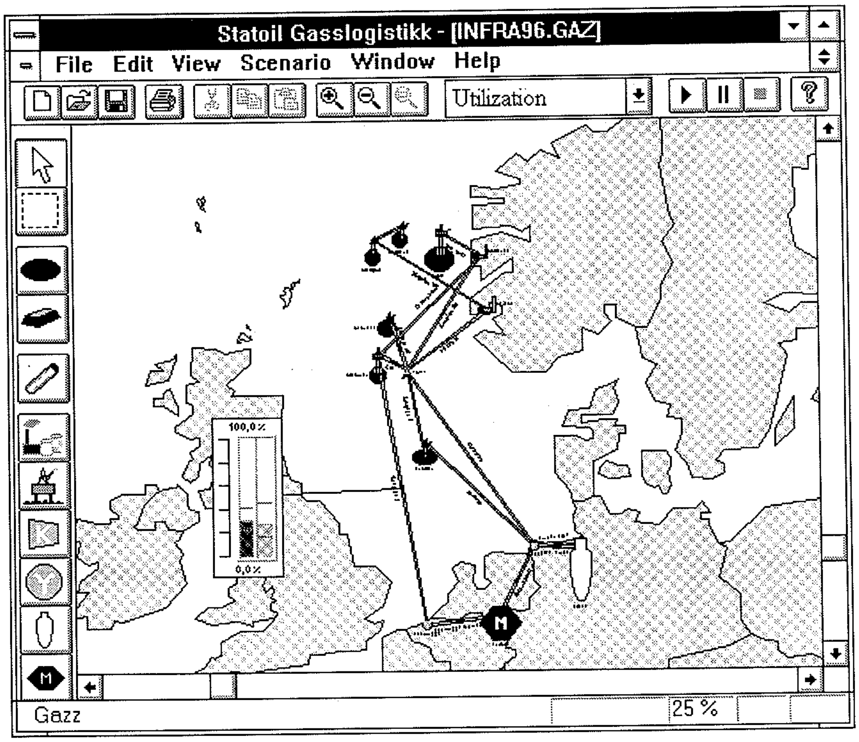

Figure 6: User in terface of gas micro world

Running a simulation

The user can specify time horizon for the simulation from the Scenario menu. In addition a life cycle can be assigned to each object (for example point in time when a pipe line is brought into operation).

When the user presses the play button the simulation starts.

Investigating simulation output

During and after simulation the user can observe model behavior in reports that may be created in the diagram (map area) or in separate windows. In Figure 6 a bar is displaying the current utilization (% of maximum capacity) of a pipe during the current run (left hand bar) and a reference run (right hand bar). Minimum, maximum, and average values can be displayed together with current values. This makes it possible to identify bottle necks, for example elements where the maximum capacity have been reached, or elements that have unnecessary large capacity (potential cost reductions). Oscillations caused by varying demands between warm and cold seasons are illustrated in time graphs that can be opened from any object. Global measures of produced volume, income and expenses are also available.

Creation of the micro world itself

The micro world was designed as a two-layer application with an accumulator–flow model at the bottom and an object model at the top. The accumulator–flow model was created in POWERSIM® (Modell-Data 1994) and the object model in C++. POWERSIM's API interface was used to link the two layers.

The core part of the simulation model is displayed in Figure 7. Arrays are used to provide a pool of reservoirs, wells, pipes, etc. that the object layer may draw from when the user creates new objects.

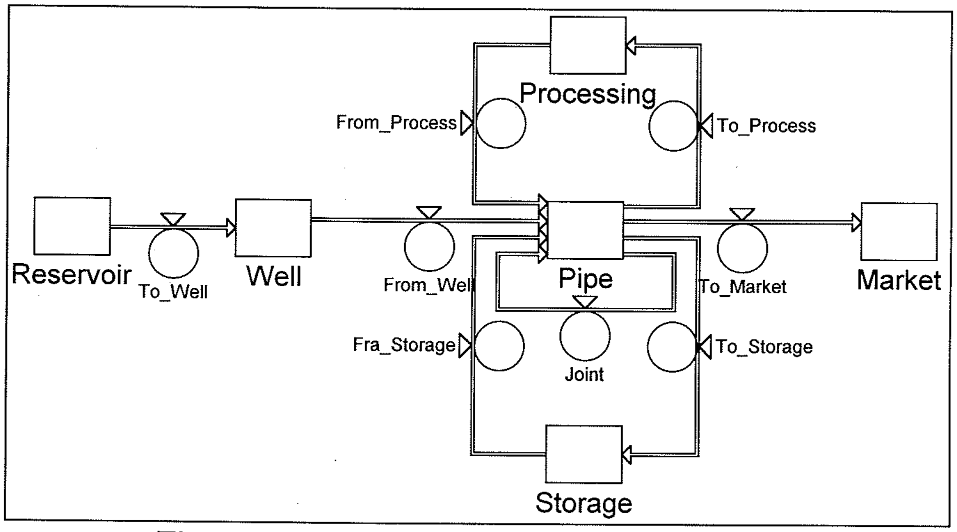

Figure 7: POWERSIM model of gas micro world

Project organization

The planner is the target user of the gas micro world. Using experimental simulation, alternative approaches can be analyzed and compared.

In order to create a micro world that is a valid mapping of the real gas world, people at different levels of the organization were involved. The expertise of researchers and experts within the company was necessary to define the objects that would be used to create the model. Simplifications and validations were done using available data from years of operation in the North Sea.

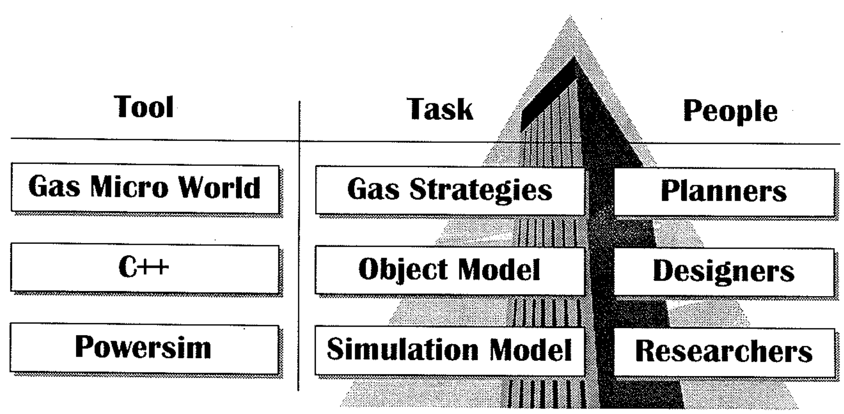

Figure 8: Teaming up people

Project evaluation

A program for testing and evaluating the gas micro world is in progress. Testing with real users is the fundamental part of the evaluation. A "thinking aloud protocol" combined with "cameraless videotaping" (recording of display images) of user interaction give valuable information. There is little doubt that the system gives planners real modeling power. At the time of writing, the data analyses are not conclusive. More field testing and data collection will be done, and the results will made available by the authors when the project is completed.

References

Andersen D.F., Chung, I.J., Richardson, G. & Stewart, T. (1990). *Issues in designing interactive games based on system dynamics models*. Organizational learning. In D.F. Anderson, G. Richardson & J.D. Sterman (Eds.), Proceedings of the 1990 International System Dynamics Conference (Vol. I, pp. 31-46). Lincoln: The System Dynamics Society.

Argyris, C., and Schøn D (1978), *Organizational Learning: A Theory of Action Perspective*, Reading, Mass.: Addison - Wesley.

Bakken, B.E., Gould, J.M. & Kim, D.H. 1992. *Management flight simulators and organizational learning: some experimental evidence*. European Journal of Operational Resarch, 59, 167-182.

Bakken, B.E 1993. *Learning and transfer of understanding in dynamic decision environments*. Unpublished PhD Thesis, Cambridge: Sloan School of Management, Massachusetts Institute of Technology.

Bean, M.P., Diehl, E.W. & Kreutzer, D.P. 1992. *Strategy simulation and scenario planning: An application of generic system structures*. Organizational learning. In J.A.M. Vennix, J. Faber, W.J. Scheper, & A.T. Takkenberg (Eds.), Proceedings of the 1992 International System Dynamics Conference (pp. 69-77). Lincoln: The System Dynamics Society.

Brown, J. S. 1989. *Toward a new epistemology for learning*, In C. Frasson and J.Gauthier (Eds.), Intelligent Tutoring System, Norwood: Ables.

Collins, A. & Brown, J. S. 1988. *The computer as a tool for learning through reflection*. In H.Mandl, A.M. Lesgold (Eds.), Learning issues for Intelligent Tutoring Systems (pp.1-18). New York: Springer Verlag.

Davidsen, P 1994. In Morecroft, J.D.W & Sterman, J.D (Eds), *Modeling For Learning Organizations* (pp. 301-317). Portland, Productivity Press.

Elsom-Cook, M. 1990. *Guided discovery tutoring; A framework for ICAI research*. London:Paul Chapman.

Faryniarz, J.V. & Lockwood, L.G. 1992. *Effectiveness of microcomputer simulations in stimulating environmental problem solving by community college students*. Journal of Research in Science Teaching, 29, 453-470.

Forrester, J.W. 1961. *Industrial Dynamics*. Cambridge, MA: MIT Press.

Foster, R. N. 1986. Innovation – the Attacker's Advantage. Summit Books, New York

Goodyear, P. 1991. *A knowledge-based approach to supporting the use of simulation programs*. Computers in Education, 16, No. 1, 99-103.

van Joolingen W. 1993. *Understanding and facilitating discovery learning in computer-based simulation environments simulation*. PhD Thesis. Eindhoven: Eindhoven University of Technology.

Kim, D 1990. *Total quality and system dynamics: complementary approaches to organizational learning*. In D.F. Anderson, G. Richardson & J.D. Sterman (Eds.), System Dynamics '90 (Vol. II, pp. 539-534).

Kreutzer, B., Kreutzer, D.P. & Gould, J.M. 1992. *The Quahog oil production simulator: A case study of the rapid development of a management flight simulator for training*. In J.A.M. Vennix, J. Faber, W.J. Scheper, & A.T. Takkenberg (Eds.), Proceedings of the 1992 International System Dynamics Conference. (Vol. I: pp. 329-336). Lincoln: The System Dynamics Society.

Lajoie, S.P., Lesgold, A. 1988. *Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship*. Man Mediated Learning, 3, 7-28.

Lane,D.C. 1994. In Morecroft, J.D.W & Sterman, J.D (Eds), *Modeling For Learning Organizations* (pp. 85-117). Portland, Productivity Press.

ModellData AS 1994. *POWERSIM 2.0 User's Guide and Reference*

Morecroft, J.D.W. 1992. *Design of a learning environment. The Oil Producers' Microworld.* Organizational Learning. In J.A.M. Vennix, J. Faber, W.J. Scheper, & A.T. Takkenberg (Eds.), Proceedings of the 1990 International System Dynamics Conference (Vol. I, pp.465-474). Lincoln: The System Dynamics Society.

Myrtveit, M. 1995. *Feeding your models with real business data.* Proceeding International System Dynamics Conference

Papert, S 1980. *Mindstorms*, Harvester, Brighton.

Peterson, S 1994. In Morecroft, J.D.W & Sterman, J.D (Eds), *Modeling For Learning Organizations* (pp. 291-300). Portland, Productivity Press.

Sellman,J. 1992. *Learning Environment.* Unpublished PhD Thesis. Milton Keynes: The Open University.

Simons, K. 1990. *New Technologies in Simulation Games*. In D.F. Anderson, G. Richardson & J.D. Sterman (Eds.), System Dynamics '90 (pp.1047-1058). Lincoln: The System Dynamics Society.

Senge, P & Sterman, J.D. 1994. In Morecroft, J.D.W & Sterman, J.D (Eds), *Modeling For Learning organizations* (pp. 195-219) Portland, Productivity Press.