Collaborative Modeling in Group Learning Environments

Lars Bollen, H. Ulrich Hoppe, Marcelo Milrad, Niels Pinkwart

Institute for Computer Science and Interactive Systems
Faculty of Engineering, University of Duisburg
47048 Duisburg, Germany
Phone: (+49) 203 379-1403
{bollen, hoppe, pinkwart}@informatik.uni-duisburg.de

Växjö University,
School of Mathematics and Systems Engineering
351 95, Växjö, Sweden
Phone: (+46) 70 782 01 34
marcelo.milrad@framkom.se

ABSTRACT

In this paper we present an interactive tool for modeling and simulation as a basis for learning with models and learning by modeling. This tool, called Cool Modes, is capable of integrating several modeling languages in one workspace. There are language plug-ins, e.g., for System Dynamics, Petri Nets, and argumentation graphs. Furthermore, features like handwritten notes or mathematical graphs are integrated. Collaboration and group learning is facilitated in the form of synchronous work with shared and private workspaces in a distributed computing environment.

Keywords: *cooperative systems, visual language environments, System Dynamics*

1. Introduction

Interactive tools for modeling and simulation (Maier and Größler 2000) are gaining increasing importance as means to explore, comprehend, learn and communicate complex ideas. In a variety of learning contexts ranging from exploratory to task-orientated environments, students benefit from building and using simulations thus developing both procedural skills and conceptual competencies (Alessi 2000). A dimension of particular interest in the educational use of computer simulations is whether and when one learns by building simulations or by using existing simulations (Spector 2000). The System Dynamics community has historically been more oriented towards learning by creating simulation models, although some designers of System Dynamics based educational environments have tried to widen the scope by addressing a variety of different learning situations and requirements (Forrester 1985; Spector 2000). The System Dynamics community is committed to and believes in the value of using System Dynamics in order to improve the understanding of complex dynamic systems (Davidsen 1996; Sterman 1994). This general commitment allows for both

learning with models and learning by modeling. How can these modeling and simulation tools be used to facilitate learning about complex domains?

In accordance with recent advances in research on learning and instruction, there are attempts to provide increasingly meaningful learning experiences. In complex domains such experiences include the ability to construct models in addition to using models for experimentation. Recently, Milrad, Spector and Davidsen (in press) have suggested an approach called "Model Facilitated Learning" (MFL) in combination with instructional design principles. Key aspects of this design framework include the use of modeling tools, construction kits and System Dynamics simulations to provide multiple representations to help students in developing an understanding of problems in situations that comprise many interrelated components which are subject to change over time and often involve ill-defined aspects. MFL distinguishes learning by modeling from learning with models and suggests when and why each approach is most likely to be appropriate. In addition, MFL emphasis the notion of socially situated learning experiences threads throughout elaborated learning sequences. Here, the notion of social situatedness extends to the idea of collaborative modeling.

On the other hand, collaborative learning is a major trend in the educational computing community. A first suggestion of how to support collaboration with modeling tools in "discovery learning" has recently been made by van Joolingen (van Joolingen 2000; Löhner and Joolingen 2002). In the construction of models using Systems Dynamics tools, learners engage in cognitive and social processes that promote collaborative knowledge building. Rouwette, Vennix and Thijsson (2000) argue that a collaborative approach to model and policy design is effective to foster learning and understanding. Accordingly, we see a new challenge in providing modeling tools in a collaborative, distributed computing framework. This is typically achieved through shared workspace environments which allow a group of learners to synchronously co-construct and elaborate external representations. Concerning the domain content of these representations, two poles can be found: on the one hand, System Dynamics models or Petri Nets provide a complete semantic definition of all objects and thus allow for running the models as simulations. In contrast, less specific systems like Belvedere (Suthers et al. 1995) do not interpret the semantic content of the objects but the rhetorical or argumentative types and relations between objects (e.g. "hypothesis", "conclusion"). The system is aware of the developed argumentation structure and points out missing relations via a support agent.

The CardBoard environment (Hoppe et al. 2000b) allows for creating "collaborative visual languages" by parameterizing a general shared workspace environment. The particular language profile specifies the syntax of the respective language, i.e. the given set of relations, their argument slots, and the basic object types. To add semantics in terms of domain models or knowledge bases, an interface is provided that transfers actions from the visual language environment to the semantic plug-in component (Mühlenbrock, Tewissen and Hoppe 1997). This architecture allows for flexibly defining semantically enriched tools, such as, e.g., a cooperative editor and simulator for Petri Nets (Wagler 1998). In our current work reported in this paper we draw on the CardBoard experience.

2. The Cool Modes framework

The Cool Modes (*CO*llaborative *O*pen *L*earning and *MODE*lling *S*ystem; Pinkwart, Hoppe and Gaßner 2001) environment re-implements and extends the CardBoard framework in the following aspects:

- The possibility of mixing different types of languages or representations, ranging form free-hand drawings over concept maps to semantically defined modelling languages (Petri Nets, System Dynamics), in one workspace.

- The provision of multiple "language palettes" to chose from; the palettes themselves being the "language plug-in" to be provided by application programmers.

- An improved underlying communication mechanism (Java MatchMaker TNG).

Cool Modes allows the use of multiple workspaces represented in different windows which can be arranged freely. Each workspace consists of a number of transparent layers which can contain "solid" objects like, e.g., handwriting strokes, images and other media types. Four predefined layers with different functionality exist by default - one for a background image, one for hand-written annotations and two for other objects - more can be dynamically added.

While the workspaces contain the results of the user's work and interaction with the system or other users, the basic elements available for this interaction are defined in "palettes". These can be dynamically added and removed and are the basic means of extending the system. Yet, some standard palettes useful for any domain are predefined: Cool Modes offers a "handwriting" palette allowing the user to directly annotate anything within the workspaces. This is useful e.g. when using the system with a pen-based input device or with an electronic whiteboard. The second more general palette consists of different patterns for discussion and argumentation support like "question" or "comment". The elements of this palette are designed to support the users in discussing their current work and structuring their argumentation. Other palettes containing more domain-specific semantics are available, too: there is a palette for modeling Petri Nets and another palette for creating System Dynamics models. Both palettes not only provide support for modeling and visualization, the models created can even be simulated; the user can "run" these models. Some more palettes are under construction at the moment; the topics of these palettes will be "simulating stochastic experiments" and "primary school mathematics".

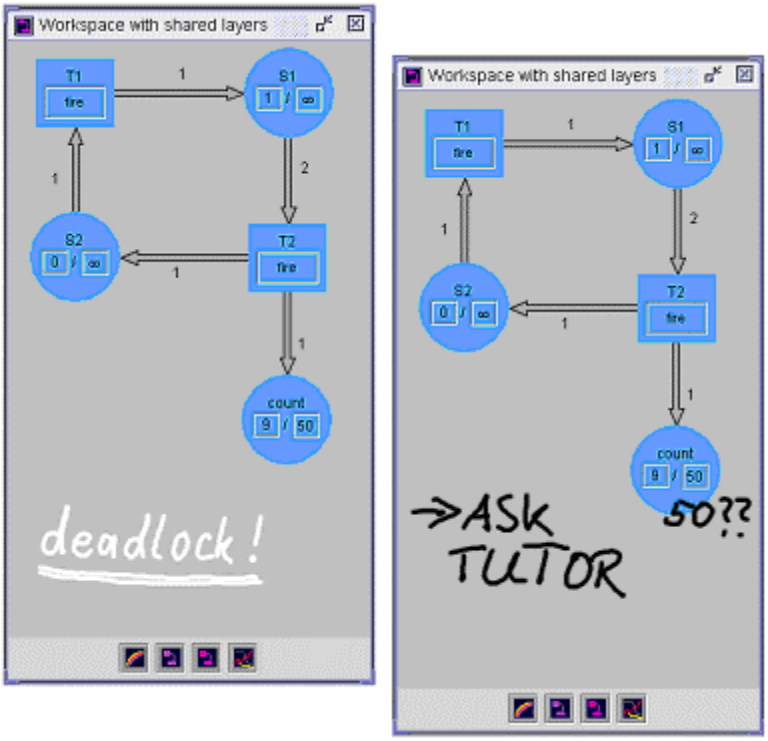

Figure 1: A Petri Net in a shared workspace with private annotations

As mentioned, the collaboration support integrated in Cool Modes relies on the provision of synchronously shareable representations. Technically, this is achieved through the MatchMaker communication server (Tewissen et al. 2000; Jansen, Pinkwart and Tewissen 2001) offering a replicated architecture, partial synchronization features and dynamic synchronization. According to the system structure with its workspaces and layers, the synchronization of objects is flexibly possible, e.g. it is possible to share the complete application or to share only specific objects, like workspaces, layers or even single objects (see figure 1).

In our vision of future application (and already starting in our own practice), we see multi-functional and multi-representational tools such as Cool Modes as digital, active extensions of the chalkboard and paper & pencil. The tools should ideally be used in networked ubiquitous and potentially mobile computing environments to support modelling, interactive presentation and group discussion in a variety of educational scenarios, including traditional lectures (presentation) as well as tutorials and collaborative work in small groups.

The use of the tools should not be conceived as a special learning mode defined by "working on or with the computer" but as an instrumental extension of a specific scenario. One of the most frequent learning scenarios is still the classroom. Our idea of a "computer-integrated classroom" has been practically elaborated and put into practice in the European NIMIS project (Hoppe et al. 2000a). The most evident and concrete result of NIMIS is a classroom installation which features special hardware such as an interactive whiteboard and pen-based tablets embedded in the pupils desks in a networked environment with educationally motivated groupware functions. Although the target group of the NIMIS project were "early learners", certainly too young for

using symbolic modeling tools, we are now extending this approach to elder age groups including high school and academic environments.

Figure 2: The NIMIS classroom

In the development of these collaborative learning scenarions with ubiquitous and distributed computing elements, we have formulated and applied the following principles:

- provide uniform access to multiple representations of media and use a variety of information sources;
- do not let the technology "get in the way" but facilitate existing classroom procedures;
- do not let the educational scenario be determined by the use of a computer, but let interactive digital media a "resource at hand" in the background similar to the traditional use of paper and pencil;
- exploit the value added from being able to easily replicate, distribute, and re-use externalized learning results in a networked digital environment.

3. System Dynamics in Cool Modes

Recently, Cool Modes has been extended with a language palette for System Dynamics modeling (Bollen 2001). Figure 3 shows an example of a model built with this *"DynaBoard"*, the particular palette in Cool Modes providing nodes and edges for building System Dynamics models. The DynaBoard palette, which is illustrated at the right side of figure 3, basically consists of three types of nodes and two different types of edges. There is a stock node (rectangle) which can hold and collect values, there is a constant node (circle) which represents a fixed value and there is a rate node (rhomb) which can be used to control the flow between two stock nodes and to calculate intermediate results. A rate node changes its color to show if it is used to calculate intermediate values (light green) or if it is used to control a flow (dark green).

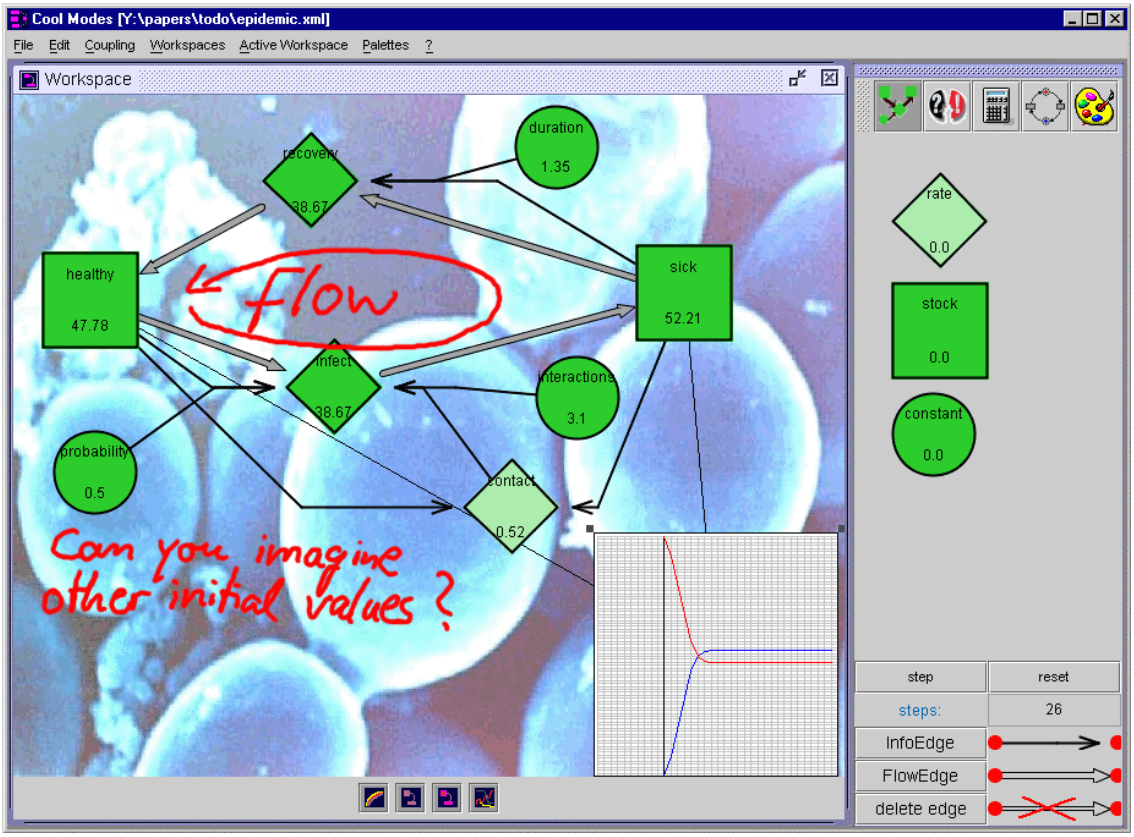

Figure 3: A System Dynamics model with hand-written notes and the DynaBoard palette

There are two types of edges: "InfoEdges" and "FlowEdges". As the names indicate, a FlowEdge is used to actually shift values from one stock to another, according to the value calculated in the connected rate node. An InfoEdge is used to provide read access to values of other nodes (without changing these). In contrast to other System Dynamics tools like Stella (High Performance Systems, Inc.) or Vensim (Ventana Systems, Inc.), a rate cannot have a *graphical* or a *list*-function, yet. Nevertheless, all basic generic

structures like various types of feedback loops, S-shaped growth or oscillating systems can be modeled and simulated.

The example shown figure 3 is a modified version of a model taken from Glick and Duhon (1994) and simulates the expansion of a disease. Two stocks are used to represent the sick and the healthy people. The underlying mathematical formulas cannot be seen in this screenshot, but they can be integrated in a model in a rather simple way: You just open a dialog box and type in the formula using the names of the nodes which are connected via InfoEdges. Done so, the formula for e.g. the „recovery"-node could be: *sick / duration*. That means: the more sick people the more can recover and the shorter the duration of the illness the more can recover. The simulation engine implicitly generates a difference equation to calculate the new values of all all nodes step by step. The engine substracts the outgoing flow and adds the incoming flow to the stock according to the values of the connected rates. Currently, the mathematical engine underlying the simulation is based on the Euler-Cauchy approach.

It should be underlined that we do not seek our innovation in the simulation engine but in new ways of interactively and cooperatively using and building simulation models. The System Dynamics representation has several advantages over alternative approaches such as directly coded simulation programs or Excel sheets: It allows for a "topographic view" of the model which reveals the basic components and their interrrelations without necessarily looking at calculation details. And even at the level of mathematical expressions which determine the quantitative dynamics of the model, the learner is only confronted with local dependencies in the form of difference equations. There is no need to master calculus and particularly integration methods to derive global model behavior from equations such as *healthy(t) = healthy(t-dt)+(recovery-infect)*dt*.

The value added of Cool Modes for System Dynamics modeling lies in the possibility of combining the DynaBoard palette and language with other palettes mentioned above and and in the possibility of collaboratively creating, inspecting, and running System Dynamics models. This is combined with general pen-based annotation or "whiteboard" facilities which are used in the same way as paper and pencil, but also potentially in collaborative mode between several computers. Thus, Cool Modes provides new room and opportunities for collaborative modeling in group learning environments. In more detail, we can distinguish the following aspects of computer support in collaborative modeling:

- Several students can share a running model by synchronizing their simulation environments. In Cool Modes this is not achievd by "windows sharing" but by synchronizing potentially autonomous environments in a so-called "replicated architecture".

- In the same way, also model building can be shared. Here not only the modelling language but also handwriting (sketches) may be used.

- Simulations are analyzed to generate hypotheses about the global behavior of systems. To do this in the form of group work, free-hand sketches as well as

argumentation graphs and mathematical tools (function plots, tables, etc.) are useful tools.

- Data can be collected in a distributed working mode with different parameters. Shared workspaces allow for gathering data from different groups.

- Group work can be supervised by sharing the environment with a distant tutor.

Cool Modes facilitates all these collaborative modes in specifc ways. Imagine the following scenario: *There is a group of learners in a room; each learner has a computer of his own and runs Cool Modes. They start to discuss a topic chosen by the teacher and use Cool Modes to support the discussion and to make (shared or private) annotations. Having agreed on the basic constrains of the problem discussed, they can arrange themselves in sub-groups, each group starting a MatchMaker session of its own. The learners can now try to show or even solve the problem by building a model using the various palettes available in Cool Modes – especially the DynaBoard palette. The elements of each palette can be used in a shared workspace, which means they can try to construct a model collaboratively. By the time problems occur, the learners (or the teacher) can make annotations to their models – either handwritten or by using the discussion palette. Even more, due to the capabilities of the MatchMaker server, a teacher could join each session at any time to have a look at the work or to assist.*

Figure 4 shows an example of a more complex model developed with students in a seminar on using computational modeling techniques in science education. The model simulates the dynamics of a "water rocket", i.e., a rocket that is propelled by water emitted from a vessel which contains both the water and the pressurized air (Schecker 1993). The model is comprised of several parts which can be elaborated quite independently:

- (Marked in red) The velocity with which the water is emitted through the nozzle of the vessel is calculated using Bernoulli's law for laminar flows and the basic theorem for adiabatic expansion of an an ideal gas. This is probably the most difficult calculation.

- (Marked in yellow) The actual mass of the rocket is calculated considering the loss of water. This allows for using the "rocket equation" $m_R * dv_R = dm_W * v_W$ (m_R: rocket mass, v_R: rocket velocity, m_W: mass of the expelled water, v_W: emission velocity) to calculate the increment in the rocket's velocity, i.e., its acceleration.

- (Upper part) The acceleration originating from the thrust is superimposed with gravity and friction to calculate the height increment and ultimately the actual height of the rocket. This model was successfully tested with real experiments.

Modeling environments, like the one presented in this section, provide the functionality to support the collaborative and constructive aspects of learning while using System Dynamics models and simulations. According to Spector (2000), collaboration and

construction in reasoning with and about models appears to be an important factor for enhancing learning outcomes.

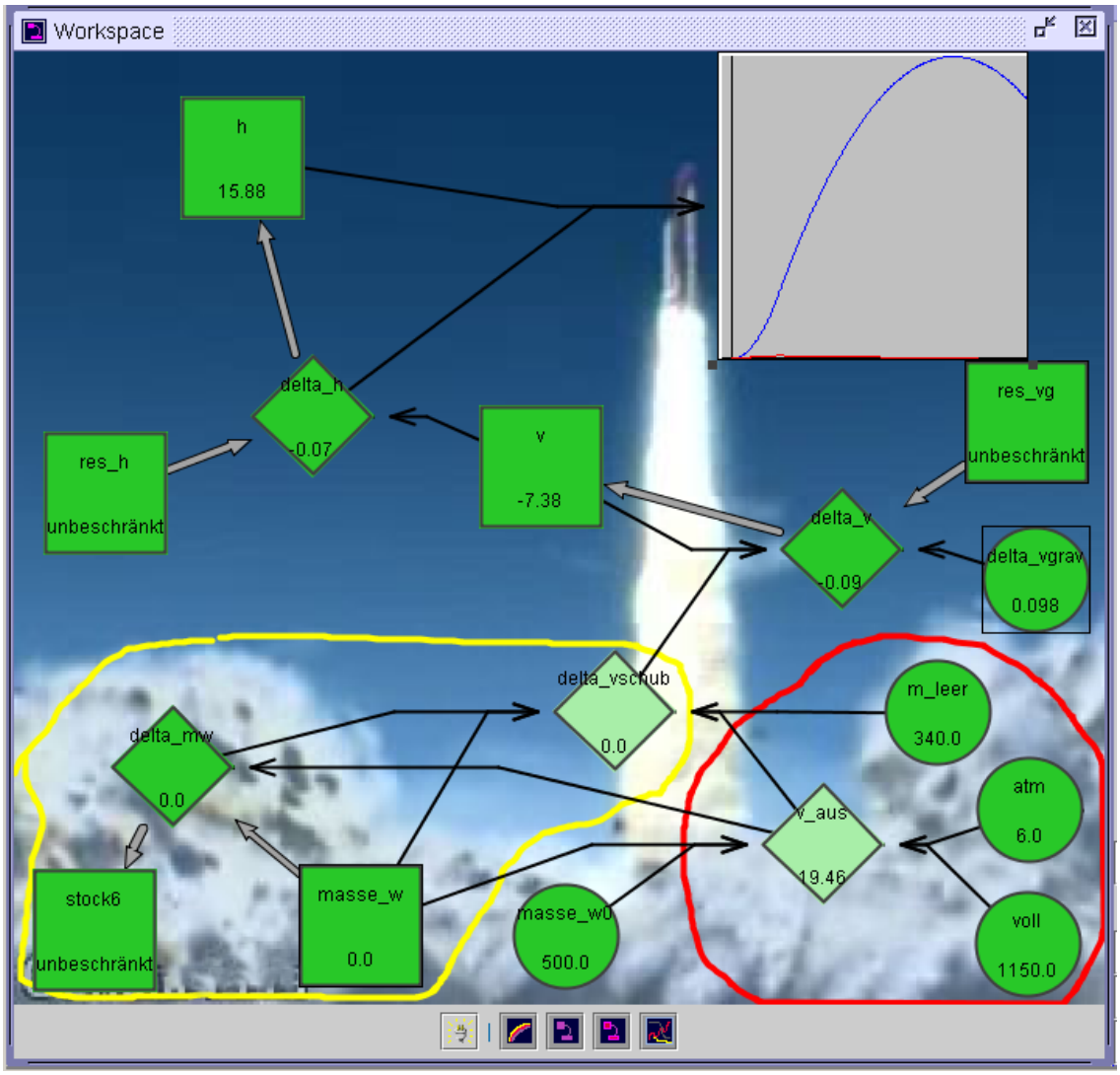

Figure 4: Simulating the flight of a water-rocket

4. Perspectives

Milrad, Spector and Davidsen (in press) suggest that learning by modeling and learning with models should be combined for the design of meaningful learning activities to support complex learning. Thus, learners need to generate hypotheses, design experiments, analyse data, predict results and rethink their hypotheses in order to collaboratively construct knowledge about the domain they are studying. Spector (2000) refers to the possible advantages of using System Dynamics simulations when peer-peer discussion and collaboration are effectively supported. There are some indications that most of the learning appears to occur in the small group discussions and not only in the interaction or series of interactions with the simulation model.

New modeling tools, as the one we described in this paper, provide a new arena for modeling and collaboration. Cool Modes and DynaBoard present some interesting features to support learning including:

- The importance of being able to represent multiple perspectives of a problem;
- The support of learning as a shared, collaborative activity - particularly in the context of bridging these multiple perspectives;
- Alternative ways to support interaction, collaboration and reflection both "around the System Dynamics simulation" as well as "beyond the System Dynamic simulation".

Another important and interesting perspective is the possibility of making models even more distributed by having different parts of one large model created on different computers. Two or more learners can create (at first glance…) independent parts of a model in their application. By coupling specific nodes (preferably stock- or rate-nodes) which are situated in both parts of the complete model, the parts can communicate which each other and will have effects on the model on the other machine(s). Using these types of "shared nodes", a new form of collaboration can be created and unique learning scenarios are imaginable.

We plan to continue the development and evaluation of these tools within the framework of a new European project we will start during the spring this year. As we will continue to conduct more research, we will gain a richer understanding concerning the potential of using these innovative tools for improving learning using System Dynamics simulations.

6. References

Alessi, S. 2000. Building versus using simulations. In *J. M. Spector and T. M. Anderson, eds. Integrated and holistic perspectives on learning, instruction and technology: Understanding complexity.* Dordrecht: Kluwer.

Bollen, L. 2001. Integration einer visuellen Modellierungssprache in eine cooperative Diskussionsumgebung für den naturwissenschaftlichen Unterricht. (Integration of a visual modeling language in a cooperativ discussion environment.) Unpublished Master Thesis, University of Duisburg, Dept. of Mathematics / Computer Science.

Davidsen, P. I. 1996. Educational features of the System Dynamics approach to modelling and simulation. *Journal of Structural Learning 12 (4).*

Forrester, J. W. (1985). The model versus a modeling process. *System Dynamics Review 1 (1).*

Glick, M., and T. Duhon. 1994. Generic Structures: S-shaped Growth I. Prepared for the *MIT System Dynamics in Education Project.* Massachusetts Institute of Technology: http://sysdyn.mit.edu

High Performance Systems, Inc.: http://www.hps-inc.com

Hoppe, H. U., A. Lingnau, I. Machado, A. Paiva, R. Prada and F. Tewissen. 2000a. Supporting collaborative activities in computer-integrated classrooms – the NIMIS approach. In *Proceedings of 6th International Workshop on Groupware, CRIWG 2000, Madeira, Portugal, 18 - 20 October 2000*, IEEE CS Press.

Hoppe, H. U., K. Gaßner, M. Mühlenbrock and F. Tewissen. 2000b. Distributed Visual Language Environments for Cooperation and Learning: Application and Intelligent Support. *Group Decision and Negotiation 9: 205-220.*

Jansen, M., N. Pinkwart and F. Tewissen. 2001. MatchMaker - Flexible Synchronisation von Java-Anwendungen. In *R. Klinkenberg, S. Rüping, A. Fick, N. Henze, C. Herzog, R. Molitor and O. Schröder, eds. LLWA 01 - Tagungsband der GI-Workshopwoche "Lernen-Lehren-Wissen-Adaptivität". Forschungsbericht 763, Oktober 2001.* Universität Dortmund, Germany.

Joolingen, W. R., van. 2000. Designing for collaborative discovery learning. In *G. Gauthier, C. Frasson and K. VanLehn , eds. Intelligent Tutoring systems.* Berlin: Springer.

Löhner, S., and W. R. van Joolingen. 2002. The effect of representations on communication and product during collaborative modeling. In *Proceedings of CSCL 2002.* Hillsdale, NJ: Lawrence Erlbaum.

Maier, F., and A. Größler. 2000. What are we talking about? A taxonomy of computer simulations to support learning. *System Dynamics Review 16(2).*

Milrad, M., M. Spector and P. Davidesen. (in Press). Model Facilitated Learning. Book chapter to appear in *"E-Learning: Technology and the Development of Learning and Teaching".* Kogan Page Publishers UK.

Mühlenbrock, M., F. Tewissen and H. U. Hoppe. 1997. A Framework System for Intelligent Support in Open Distributed Learning Environments. In *B. du Boulay and R. Mizoguchi, eds. Artificial Intelligence in Education: Knowledge and media in learning systems, Proceedings of the 8th International Conference on Artificial Intelligence in Education , AIED-97.* Amsterdam: IOS Press.

Pinkwart, N., H. U. Hoppe and K. Gassner. 2001. Integration of Domain-specific Elements into Visual Language Based Collaborative Environments. *In Proceedings of CRIWG 2001 (International Workshop on Groupware). Darmstadt, October 2001*. IEEE Press.

Rouwette, E. A. J. A., J. A. M. Vennix and C. M. Thijssen. 2000. Group model building: A decision room approach. *Simulation & Gaming 31(3).*

Schecker, H.P. 1993. The Didactic Potential of Computer Aided Modeling for Physics Education. In *D. L. Ferguson, ed. Advanced Educational Technologies for Mathematics and Science.* Berlin: Springer

Spector, J. M. 2000. System Dynamics and interactive learning environments: Lessons learned and implications for the future. *Simulation & Gaming 31(4).*

Sterman, J. 1994. Learning in and about complex systems. *Systems Dynamics Review 10 (2-3).*

Suthers, D. D., A. Weiner, J. Connelly and M. Paolucci. 1995. Belevedere: Engaging students in critical discussion of science and public policiy issues. In *Proceedings of the World Conference on Artificial Intelligence in Education.* Washington, DC: American Association for the Advancement of Computation in Education.

Tewissen, F., N. Baloian, H. U. Hoppe and Reimberg, E. 2000. "MatchMaker" Synchronising Objects in Replicated Software-Architectures". In *Proceedings of 6th International Workshop on Groupware, CRIWG 2000 Madeira, Portugal, 18 - 20 October 2000,* IEEE CS Press.

Ventana Systems, Inc.: http://www.vensim.com/

Wagler, F. 1998. Erstellung eines Petrinetz-basierten Steuerungsmechanismus zur Behandlung von Turn-Taking-Regeln in Gruppen-interaktiven Spielen. Unpublished Master Thesis, University of Duisburg, Dept. of Mathematics / Computer Science.