Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

SEAMLESS INTEGRATION OF GAME AND LEARNING USING MODELING AND SIMULATION

Dennis Maciuszek

University of Rostock
Albert-Einstein-Str. 22
D-18059 Rostock, GERMANY

Martina Weicht

University of Rostock
Albert-Einstein-Str. 22
D-18059 Rostock, GERMANY

Alke Martens

University of Education Schwäbisch Gmünd
Oberbettringer Str. 200
D-73525 Schwäbisch Gmünd, GERMANY

ABSTRACT

In this paper we show our approach to make two steps towards integration of modeling and simulation and intelligent tutoring for game-based learning environments. The first step is to construct a framework for developing game-based (and also non-game-based) learning systems, which is based on fundamentals of Intelligent Tutoring System construction and on a full-fledged simulation system (JAMES II), and which allows component-based design and re-usage of existing components. The second step is to develop modeling and simulation-based learning games which seamlessly integrate game and play (immersive didactics). We show the viability of our approach in two case studies – a racing game that can be used to learn geographical facts, and a game about marine science in which school students learn about the interrelations of an ecosystem.

1 INTRODUCTION

In the first years of institutional education (starting with kindergarten, up to the first years at school), game-based learning is an integral part of pedagogical concepts: first mathematical steps are made in a playful manner, writing is a game, and biological and historical backgrounds are mediated by playful experiments and stories, see e.g. Martens, Diener, and Malo (2008). In later phases of the lifelong learning process, the game-based mediation of teaching and training content is not that obvious anymore (and seems to be somewhat suspect). Sometimes, what can be perceived to be a game-based approach comes with another label, like case-based learning, e.g., Martens (2004) and Merseth (1991), or learning in role-playing games. Application domains are for example Medicine, Business Sciences or management training.

In contrast to the lack of game-based approaches in adult teaching and training, the research field of game-based learning has increased in the last years. The reasons for this can be seen in the (still unproved) hypothesis that game-based approaches help to raise the motivation and the perseverance of the learner. However, even if this can arguable not be proved, one thing can easily be observed: many people over eighteen years old spend a lot of time and energy in playing interactive games on their computers – even if the prerequisite contains training and application of complex rule systems (like for example in *World of Warcraft*, see Rausch, Fasshauer, and Martens 2012). It would really be a nice to have if an appealing game-based learning environment, which could keep learners engaged in a similar way.

After an explication of the core terms in the following section, we describe the two steps, which we have performed in previous research in the field of game-based learning. The first step is a technical one: we

978-1-4673-4782-2/12/$31.00 ©2012 IEEE

constructed a framework for game-based (and also for non-game-based) learning systems which is rooted in Intelligent Tutoring Systems (ITSs) and which integrates a full fledged simulation system (JAMES II – JAva-based Multipurpose Environment for Simulation II). The resulting framework is sketched in Section 3. In the second step, we have developed several game-based learning systems (with and without simulations) as proof of concept. Two case studies are described in this paper in Section 4. The paper concludes with a discussion and an outlook.

2 BACKGROUND: THE TERMS

In game-based training, we often find an interplay between three main fields (see Figure 1 from Martens, Diener, and Malo 2008), which are learning, simulation and games. Training simulations, like for example used in the military domain, medicine or in business science, are used to teach and train facts using simulations. The simulation's role is to show the underlying system behavior – as realistic as possible (which does not necessarily include demanding graphics, but close to real-life models). Leaving out simulation aspects, i.e. combining learning and games, leads us usually to simple edutainment games, which are often used in primary school settings (e.g. learning how to spell a word in a game-based manner). Leaving out the training aspect leads to simulation games, which sometimes come as real simulations (in the sense of experimenting with systems consisting of models, including a temporal aspect) – sometimes, they come as games with a simulation appearance, but no core simulation functionality (see for example Martens 2008). Only in the area, where all three fields overlap, a game-based training simulation can be found, which covers all aspects: games, simulations and learning. Our approach is located at this central field.

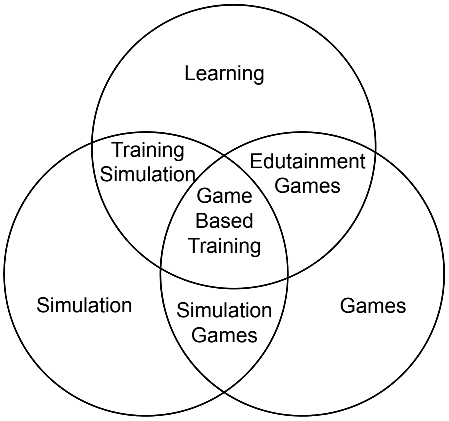


Figure 1: Interplay of pedagogy, computer science and games.

In Martens, Diener, and Malo (2008) and Martens (2008), three types of simulations for teaching and training have been distinguished: simulation of the environment, simulation of characters (i.e. non-player characters in the role-play terminology), and simulation of other models, like for example the functionality of a certain machine or equipment. These three types of simulations in teaching and training require interaction by the learners. This interaction can be realized on different levels: learner can interact directly with the models (e.g. changing parameters), they can develop models or they can just interact with the environment or with characters without beeing aware of the model or simulation character. In game-based teaching and training simulations, the simulation functionality and the models' structure are often hidden behind game interfaces, which the learner uses to indirectly interact with the models.

3 THE FRAMEWORK JABISIM

The JaBISim (Java Based Intelligent Tutoring with Simulation) framework is the result of our efforts to integrate JAMES II into JaBInT. This section will introduce each of the two frameworks and their combination.

3.1 JaBInT

Our component-based framework JaBInT – Java-Based Intelligent Tutoring – has been created and implemented as a flexible, reusable basis for Intelligent Tutoring Systems and other e-learning systems.

It consists of four semantical core components (Martens 2004; Oertel 2006):

- the *Expert (knowledge) component* containing all knowledge related to the content of the learning application along with instructional knowledge,
- the *User component* comprising knowledge about the user, the learner, such as their history, preferences, and state of knowledge
- the *Process Steering component* broking information back and forth between the other three components, and
- the *User Interface component* presenting content to the user and receiving input from them.

Each of these components consists one or more modules implementing the actual functionality and the communication between the modules. This communication is based on ports which may send (as Master or Slave ports) or receive messages based on certain communication patterns (Ruddeck, Maciuszek, Weicht, and Martens 2011).

Proof-of-concept implementations include an ITS for training chemical nomenclature (Maciuszek, Ruddeck, and Martens 2010a) with both a two-dimensional Java-based interface (ChemNom) and an *Open Sim*-based 3D version (ChemNom3D). ChemisTris (*ibid.*), a game-based version of ChemNom reuses all ChemNom modules except the user interface. Using Flash instead of Java, the latter re-implements the famous Tetris game (http://www.tetris.com) adapted to chemistry. For a more sophisticated integration of games in ITS, we have conducted a literature review (Maciuszek and Martens 2011b) and introduced a concept for a fully-integrated game-based ITS architecture based on JaBInT, both for educational computer role-playing games (*ibid.*) and for educational business simulation games (Neef, Maciuszek, and Martens 2011).

The JaBInT framework stands for a highly flexible, reusable, carefully engineered framework for adaptive learning systems.

3.2 James II

The JAva-based Multipurpose Environment for Simulation II (Himmelspach and Uhrmacher 2007), James II for short, is a framework for modeling and simulation that is open to any research paradigm, application, modeling formalism, and simulation algorithm. Its use cases can be found in research in the field of modeling and simulation, in research projects that require modeling and simulation of complex models, and in teaching (both stand-alone or as backend in teaching and training systems).

Its modular design – four main parts: modeling layer, experiment layer, graphical user interface, and execution layer – allows for easy extension, exchange, and reuse of parts of the system making it a scalable, flexible base for virtually any modeling and simulation application. A plugin mechanism lets developers adapt James II to exactly their needs. They may choose from more than 500 already existing plugins or create their own.

James II has been used in a variety of use cases including the development of modeling algorithms and model formalisms, in software testing, population development and demography prognosis, and in tutoring.

3.3 Integrating JaBInT and James II into JaBISim

Our efforts to create constructivist learning applications based on JaBInT led to the idea of adding interactive simulated game worlds to JaBInT-based educational games (Tausch 2011). During gameplay, the player changes simulation parameters, influencing the simulation-based visualization presented in the game as a human-in-the-loop.

The main design decision for JaBInT has been to let the process steering component act as a broker in between the other three components. Expert knowledge component, user component, and user interface component never communicate directly with one another. Instead they use communication ports to send and request information from the process steering component. This helps maintain a clean software design and allows for the exchange of modules.

The same principle holds for any extension of JaBInT, like in this case for adding James II. Therefore, a dedicated module was added to the expert knowledge component. It provides an interface to James II using its API, through which it exchanges simulation parameters and results. James II "feeds" the expert knowledge component. The process steering component also received a new module for coordinating simulation needs and sychronizing simulation and game. Whenever there is a need for new simulation data during the game, the "synchronization" module within the process steering component requests so from the "simulated game world" module in the expert knowledge component. This module contacts James II via its API, sends new simulation parameters and receives the results from James II, passing them on to the synchronization module: Figure 2. The figure already includes the case of Section 4.1, a *Flash*-based racing game connected to JaBInT via the *JFlashPlayer* middleware.

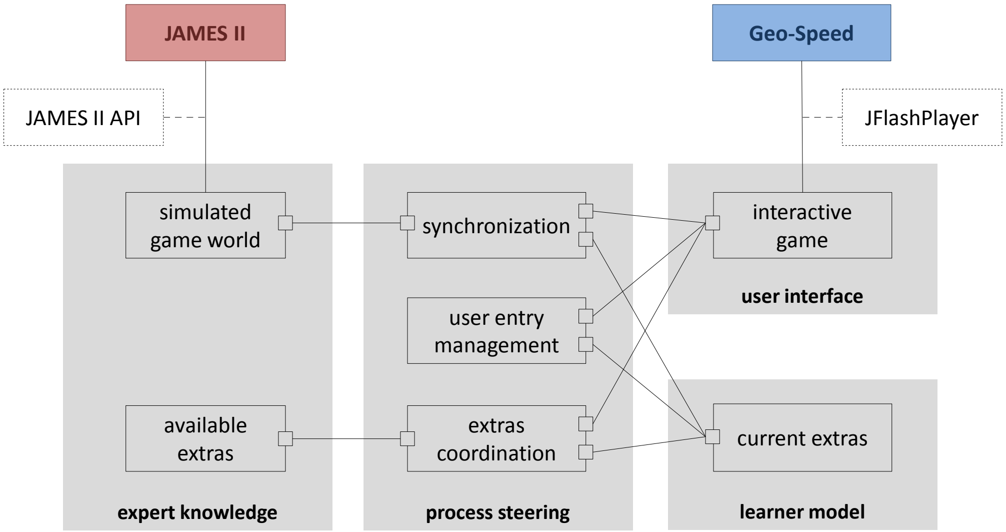


Figure 2: *JaBISim* integrating James II into JaBInT.

4 CASE STUDIES

4.1 Case: Geo-Speed

Contemporary off-the-shelf video games for education tend to fail at integrating the educational content seamlessly in gameplay (Maciuszek and Martens 2011b). For instance, the first two installments in the German-language *Genius* series (developed by RadonLabs, published by Cornelsen) are in essence construction and management simulation games interrupted by traditional e-learning exercises on science topics. The physics exercises of the first game are only connected to its construction gameplay by earning money for correct answers, which may be spent on virtual infrastructure. In the second game, it was possible to play through the construction part without even looking at the biology exercises (Jantke 2007). Immersive instruction (Bopp 2005) is a view of game design that promises to solve the learning-gameplay integration dilemma. Nowadays, designers of entertainment games employ different strategies of 'sneaking in' tutorials on operating the games into their narrative frames. For instance, the player acquires knowledge of pressing keys for moving the avatar during an interactive presentation of the background story. In educational video games, the idea would be to 'sneak in' educational content into gameplay in a similar fashion.

We tried this idea in a game on geographical facts. The traditional way would have been to realize acquisition of country positions or city names as a quiz. We thought that a more natural and appealing idea would be to have the learner participate in a virtual car rally across Europe. In its first version, *Geo-Speed* (Figure 3) tests previous knowledge through a single-choice question disguised as a customs form to be completed when entering a country. While passing through, *Geo-Speed* presents different sites and other visual information at the side of the road. When leaving a country, a reporter character poses questions on what the driver has seen. This first version of the game was implemented using *Adobe Flash* with Actionscript 3. An evaluation with school students of different ages (Tausch, Maciuszek, and Martens 2011) indicated that the idea of immersive instruction might work, at least for some. Contrary to students who had played a non-immersive version, players of the immersive game came up with creative ideas for improvements.

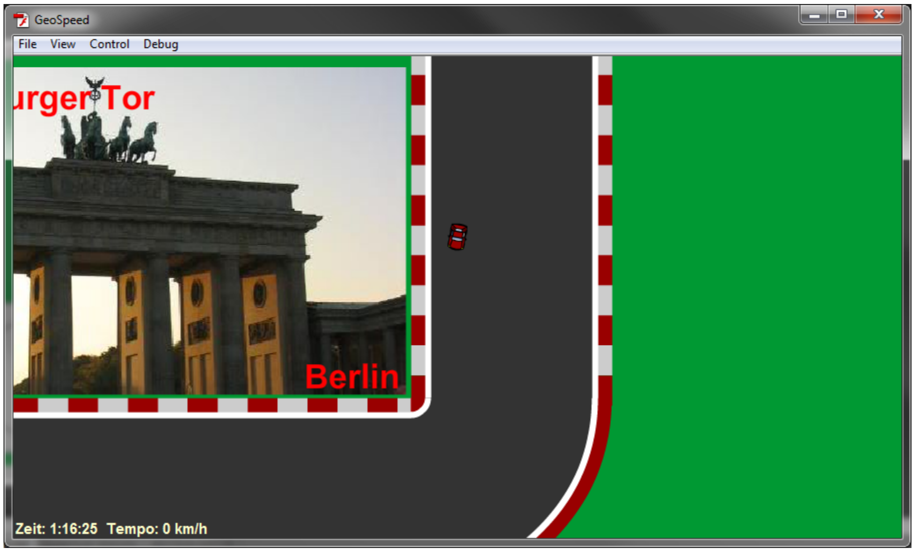
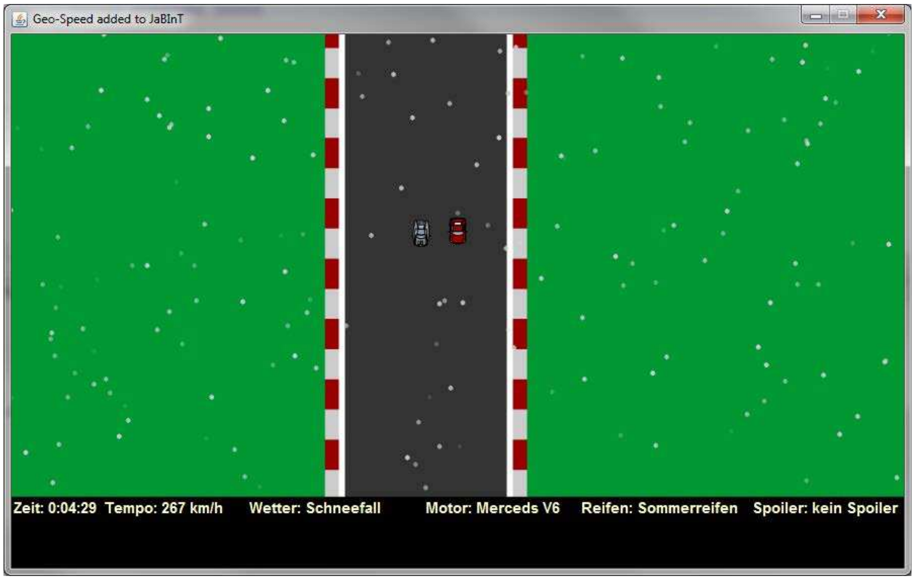


Figure 3: *Geo-Speed* without simulation.

There is more to geography, though, than memorizing facts about countries. As a next step, we developed an add-on for *Geo-Speed* with the learning goal of understanding climatic conditions over Europe. One could cover this science topic by showing pictures of climographs, adding exercises in which the player assigns them to different regions. Yet, this solution would bear as little interactivity as a regular paper work book. The only difference is that giving feedback may be automated. Instead, we wanted the learner to experience climate and weather conditions, as well as to play around with them. Experiencing long-term, spatial weather changes is not something easily achieved in a lesson in the classroom. In such cases, interactive simulations can be of help. Hence, we added a climate model to *Geo-Speed*. Changing conditions affect the roads. The player would respond to these by outfitting their car with the appropriate extras. Available extras include different tires, spoilers, and motors. Such maintenance is only available at petrol stations. Thus, the player must anticipate changing conditions in due time. In addition, correct answers to customs/reporter questions improve weather conditions, while wrong answers make things worse.

The model implemented in JAMES II is a PDEVS-based (parallel discrete-event system) grid. Each of the connected cells represents a section of the game map with certain starting values, depending on the respective region of Europe. Starting values concern temperature and the probability of rainfall. In colder temperatures, snowfall replaces rain (cf. Figure 4). By means of von Neumann neighborhood rules, changes propagate with easterly winds from right to left. Note that we simulate only weather in the strict sense. Resulting road conditions in the game change on the basis of heuristics.

To implement this simulation-based game, we made full use of the capabilities of JaBISim. We kept and extended the *Flash*-based user interface and controls, however JaBInT is now employed as a kind of middleware to steer the climate simulation in James and handle extras (cf. Figure 2). The main issue was synchronization of game time and simulation time. Would James deliver simulation results fast enough to satisfy *Geo-Speed's* real-time requirements? Could we inject desired changes fast enough? Results showed that already with a 50 by 50 grid, there was a delay in changed weather effects becoming visible. One solution would be to accelerate simulation time when needed, a strategy of scaled real-time execution.



Zeit: 0:04:29 Tempo: 267 km/h Wetter: Schneefall Motor: Merceds V6 Reifen: Sommerreifen Spoiler: kein Spoiler

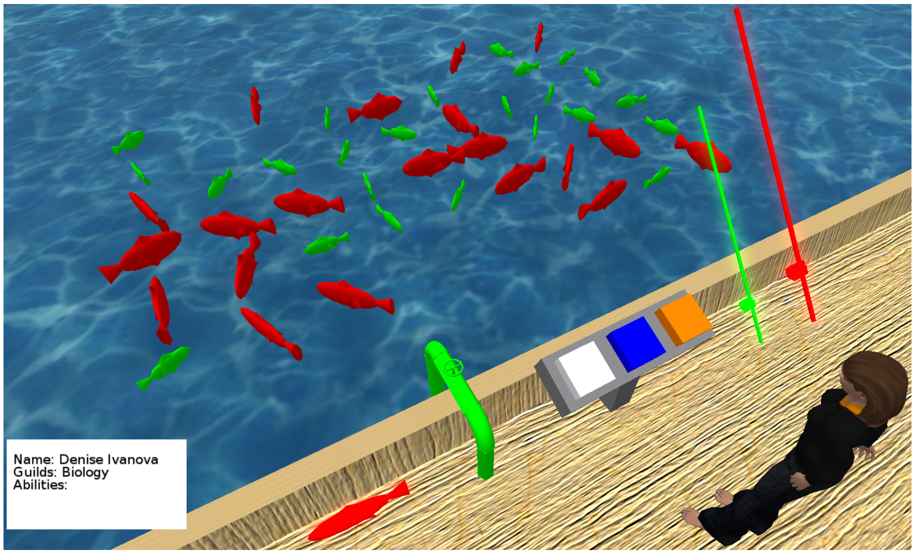
Figure 4: *Geo-Speed* with simulation.

4.2 Learning with Simulation Games

In its second version, *Geo-Speed* is an example of a simulation-based educational game implementation in which the player acts as a human-in-the-loop. Its main design flaw, though, is that player actions influencing the simulation are not meaningful in the sense of situated learning. In real life, no student giving an incorrect answer would have to fear a sudden gush of rain. Such interaction and feedback is artificially constructed, hence even in this somewhat more immersive example, educational content and gameplay actions are not well integrated. The player does not perform deliberative experimentation on the model.

To shape a truly immersive learning experience, it is not enough to 'sneak in' educational content into the story – the *Genius* games do that, too – it ought to be integrated in gameplay – in the case of simulation games in interactive simulation mechanisms. We have previously documented educational gameplay mechanisms in business simulation games (Neef, Maciuszek, and Martens 2011), science-oriented simulation games (Maciuszek and Martens 2009), and in computer role-playing games (RPGs) with simulated activities in micro-virtual worlds (Maciuszek and Martens 2011a). For instance, the activity Crafting appears in entertainment RPG series like *Drakensang*, *Dragon Age* or *Gothic*, but also in *Minecraft* – a game focusing only on that activity, and the *Harry Potter* action adventure games in which the player brews magic potions.

Crafting appears to suit engineering and science experiments in educational games. In engineering scenarios, Crafting would involve using items on other items. In science scenarios, Crafting would mean using items on the environment. We have tested both, but shall focus the following discussion on the more simulation-based interactive science experiments. A prototypical design and implementation of a Crafting exercise in the domain of biology teaches population dynamics: Figure 5. The environment includes a population of green prey fish as well as a population of red predator fish feeding on the former. The student sees fish appearing and disappearing, but no actual preying behavior. He or she may interfere with the simulation by removing green or red fish with the respective fishing rod or by adding green or red fish via the fodder tube and the thrown-in fish, respectively. The underlying model is a simple discrete-time, discrete-space approximation (Carow 1986) of the Lotka-Volterra differential equation system for population dynamics. Two publications detail technical aspects of the prototype, including the use of *OpenSim* and its connection to JaBInT (Maciuszek, Ruddeck, and Martens 2010b), and discuss instructional considerations, particularly inquiry learning via the Crafting activity (Maciuszek and Martens 2011a).



Name: Denise Ivanova
Guilds: Biology
Abilities:

Figure 5: Crafting prototype.

Catching fish as a metaphor for changing a parameter is a meaningful, immersive action in the context of population dynamics. Contrary to incorrect answers causing rain, it does not only place the learner as a human-in-the-loop, but it may actually lead to immersive 'learning in the loop'. Figure 6 shows a cycle of learning in educational simulations. An expert model gets simulated, which in turn is visualized. The student observes the phenomena to be studied, creates hypotheses, validates them by changing parameters – i.e. experimenting – and after further simulation interprets visualized results again, until he or she arrives at a reasonable explanation.

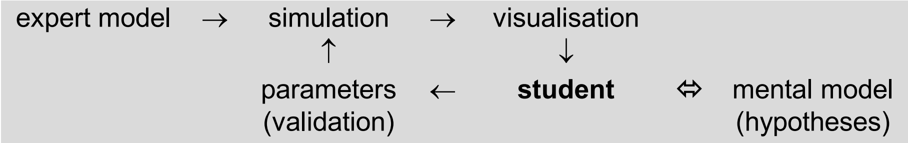


Figure 6: Learning process in educational simulations.

Hypothesis formation and validation draws on the mental model the student acquires of the simulated expert model. In educational games, we intend to support mental modeling by providing lifelike visualizations, game actions, and activities. Yet, this applies to more than visualization through game graphics and interactive gameplay processes. Simulations and their underlying models ought to exhibit realistic behavior to allow transfer of gained knowledge to real-world situations. Some educational scenarios may thus require continuous-time models, i.e. differential equation systems. James II provides modules to solve these. The question will be, again, whether the solver can respond in time for our interactive applications – and if not, whether a suitable workaround approach to synchronization can help. A further case study examines this, as outlined in the following section.

4.3 Case: Interactive Ecosystem

The EU-funded project *South Baltic WebLab* (http://www.southbalticweblab.eu) aims at raising an interest in natural science, specifically marine sciences, among high school students in their final years. In the e-learning part of the project, we are developing a series of virtual labs in which students assume the role of a scientist.

One of the experiments features an interactive marine ecosystem which the students can manipulate in order to create simple models of the portrayed biogeochemical and physical processes. Manipulation in that case means interfering with the ecological balance, e.g. by adding different amounts of nutrients. Students are to discover what degree of over-fertilization leads to eutrophication: increased photosynthesis effects excessive algae growth on the surface. This leads to a lack of oxygen on the seafloor, which kills or drives away fish populations. The design permits experimentation and hypothesis formation, and adds a tool for drawing boxes-and-arrows models of the ecosystem to support the formation of mental models and reflection on gained insights. An intelligent tutor will assist in experimenting and modeling. This science-oriented Crafting activity will later be augmented with a narrative frame, a quest in which the player is to find out who caused the ecological desaster – farmers in that scenario.

In development, we have started with a *Flash* prototype, similar to the case of *Geo-Speed*. Intelligent tutoring will be added. When it comes to simulation, the question is whether an equation solver can provide solutions fast enough. Our current model is already a massive simplification of the 'real' model normally used by our marine science partners. It contains only a few interdependent equations.

In the first version, we follow an approximation approach with table-lookup of pregenerated values. It is a mock-up simulation that does not create a realistic environment, but a reference system that is fast. In a future step, we will experiment with different implementations of the actual equations in James II.

So far, the case studies are still very much work in progress, but they indicate that – technologically – educational games may be powered by real simulations. This works – in the selected cases – on the software-engineering level for discrete-time simulations. Continuous-time simulations are under investigation using the JaBISim framework.

5 DISCUSSION AND OUTLOOK

We started our work in the field of Intelligent Tutoring Systems, but over time realized we were getting closer to the game-based teaching and training domain. Having strong roots in modeling and simulation, our first approach has been to combine the better of two worlds: Intelligent Tutoring Systems (ITSs) comprise adaptive learning technology hand in hand with a large amount of interaction and adaptive feedback based on learner performance. Simulations in teaching and training comprise elaborate models, learner interaction with systems and more realistic training situations. Moreover, from the software development perspective, it is way more elegant to have a simulated character or a simulated environment than one based on database entries. Looking closer at ITSs, they always require models of the application domain – here in the guise of knowledge models of the application domain. But if these knowledge models contain any temporal aspects, we come very close to discrete-event simulations (Martens 2006). Thus, we learned that ITSs and modeling and simulation can easily go hand in hand. Our closer look at the game-based domain revealed nothing new: also in this domain, the work with ITS structures and modeling and simulation lead us to very flexible and adaptable learning environments.

Yet, after several years of system investigation and development, we are still at the level of prototype development when it comes to real game-based learning environments. This is related to the big effort of developing appealing content related to learning and game structures. We have made progress in the *South Baltic WebLab* project, and in August 2012 an evaluation with the first groups of users took place (results will be presented in the conference talk). This far, our studies have shown the following: our framework for game-based and non-game-based teaching and training environments is very flexible and can be used for diverse types of learning systems, comprising also interactive simulations. Evaluations with users will show how they respond to the level of interaction and, from the pedagogical perspective, whether they experience any disruption of their playing by being confronted with simulation-based learning content.

ACKNOWLEDGMENTS

We would like to thank Alexander Tausch for his efforts in realizing the JaBISim vision and for developing *Geo-Speed* as a test-bed. We thank the Modeling and Simulation group at the University of Rostock who provide the James II system, especially Jan Himmelspach who has been involved in conceiving JaBISim all the way and who co-supervised Alexander's Master's thesis.

REFERENCES

Bopp, M. 2005. "Immersive Didaktik: Verdeckte Lernhilfen und Framingprozesse in Computerspielen". *kommunikation@gesellschaft* 6.

Carow, K. 1986. "Populationsökologisches Würfelspiel". *Unterricht Biologie* 112:33–38.

Himmelspach, J., and A. M. Uhrmacher. 2007, March. "Plug 'n Simulate". In *Proceedings of the 40th Annual Simulation Symposium (ANSS '07)*, edited by T. F. Znati, 137–147. Washington, DC: IEEE.

Jantke, K. P. 2007, September. "Serious Games – eine kritische Analyse". In *11. Workshop Multimedia in Bildung und Wirtschaft – Tagungsband*, edited by V. Yakimchuk, 7–14. Ilmenau.

Maciuszek, D., and A. Martens. 2009, September. "Virtuelle Labore als Simulationsspiele". In *INFORMATIK 2009 – Im Focus das Leben. Beiträge der 39. Jahrestagung der Gesellschaft für Informatik (3. Workshop Pervasive University)*, edited by S. Fischer, E. Maehle, and R. Reischuk, 1965–1979. Bonn: GI.

Maciuszek, D., and A. Martens. 2011a, October. "Computer role-playing games as an educational game genre: Activities and reflection". In *Proceedings of the 5th International Conference on Games-Based Learning (ECGBL 2011)*, edited by D. Gouscos and M. Meimaris, 368–377. Reading: Academic Publishing.

Maciuszek, D., and A. Martens. 2011b. "A reference architecture for game-based intelligent tutoring". In *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches*, edited by P. Felicia, Chapter 31, 658–682. Hershey, PA: IGI Global.

Maciuszek, D., G. Ruddeck, and A. Martens. 2010a. "Component-based development of educational games: The case of the user interface". In *Proceedings of the 4th European Conference on Games-based Learning (ECGBL 2010)*, edited by B. Meyer, 208–217. Reading, UK: Academic Publishing. Copenhagen, Denmark, 21-22 October 2010.

Maciuszek, D., G. Ruddeck, and A. Martens. 2010b, October. "Component-based development of educational games: The case of the user interface". In *Proceedings of the 4th European Conference on Games-Based Learning (ECGBL 2010)*, edited by B. Meyer, 208–217. Reading: Academic Publishing.

Martens, A. 2004. *Ein Tutoring Prozess Modell für fallbasierte Intelligente Tutoring Systeme*. Berlin: Aka.

Martens, A. 2006. "Time in the Adaptive Tutoring Process Model". In *Proc. of the 8th International Conference on Intelligent Tutoring Systems 2006*, edited by Mitsuru Ikeda, K. D. Ashley, and T.-W. Chan, Lecture Notes in Computer Science LNCS, 134–143: Springer.

Martens, A. 2008. “Simulation in Teaching and Training”. In *Encyclopedia of Information Technology Curriculum Integration*, 764–770. IGI Global.

Martens, A., H. Diener, and S. Malo. 2008. “Game-based Learning with Computers – Learning, Simulations, and Games”. *Transactions on Edutainment, LNCS* 5080:172–190.

Merseth, K. 1991. “The Early History of Case-Based Instruction”. *Journal of Teacher Education* 42 (4): 243–249.

Neef, A., D. Maciuszek, and A. Martens. 2011, July. “Mapping business simulation games to a component architecture”. In *Proceedings of the 2011 11th IEEE International Conference on Advanced Learning Technologies (ICALT 2011)*, edited by J. M. Spector and Kinshuk, 366–368: CPS/IEEE.

Oertel, M. 2006. “Entwicklung eines Frameworks für intelligente Lehr-/Lernsysteme”. Master’s thesis, University of Rostock, Germany.

Rausch, S., U. Fasshauer, and A. Martens. 2012. “Evaluation of Competency Development in WoW”. In *E-Learning and Games for Training, Education, Health and Sports. 7th International Conference, Edutainment 2012 and 3rd International Conference, GameDays 2012. Proceedings*, edited by S. Göbel, W. Müller, B. Urban, and J. Wiemeyer, 78–88.

Ruddeck, G., D. Maciuszek, M. Weicht, and A. Martens. 2011. “Communication patterns in component-based Intelligent Tutoring Systems”. *CEPIS UPGRADE* 12 (2): 26–33.

Tausch, A. 2011. “Integration einer Simulations-Engine in eine komponentenbasierte eLearning-Architektur am Beispiel der Entwicklung eines simulationsbasierten Geographie-Lernspiels”. Unpublished Master’s thesis, University of Rostock, Germany.

Tausch, A., D. Maciuszek, and A. Martens. 2011, May. “Immersive Didactics in Game-based Learning: The Geo-Speed Case Study”. In *eLearning Baltics 2011. Proceedings of the 4th International eLBa Science Conference*, edited by S. Hambach, A. Martens, and B. Urban, 123–132. Stuttgart: Fraunhofer.

AUTHOR BIOGRAPHIES

DENNIS MACIUSZEK is both a media author (Master of Arts from Stuttgart Media University, Germany) and a computer scientist (diploma with minor subject Psychology from Technical University of Braunschweig, Germany, and Licentiate of Engineering from Linköpings universitet, Sweden). From December 2008 to November 2012, he has been a teacher and researcher at the University of Rostock, Germany, working in the areas of game-based learning and virtual worlds for education. He is currently finishing his PhD at the University of Education Schwäbisch Gmünd, Germany. Dennis is interested in modeling and simulation as a basis for interactive activities in digital educational games. A collection of some of his design works and a list of academic publications can be found at his Web site, http://www.storyautor.de.

MARTINA WEICHT is a PhD student at the University of Rostock, Institute of Computer Science, Software Engineering group. She is currently working in an EU project on engaging school students into Marine Sciences, where she develops virtual lab experiments based on “real” scientific work.

ALKE MARTENS is Professor at the University of Education Schwäbisch Gmünd, leader of the research group “Computer Science and Instructional Design”, and elected vice-rector for research and international affairs. She received her PhD in Computer Science from the University of Rostock in Artificial Intelligence in the context of Intelligent Tutoring Systems. After that she was Junior Professor at the University of Rostock and leader of the research group “E-Learning and Cognitive Systems”. Her current research interests are formal methods, software engineering, modeling and simulation, teaching and training systems, and a combination thereof, e.g. in game-based learning.