Modeling, Simulation and Games

Alke Martens, Jan Himmelspach, Roland Ewald
Institute of Computer Science, University of Rostock, Germany
martens@informatik.uni-rostock.de

Abstract: Using modeling and simulation techniques in game development can look back on a comparably long history, starting in the early 1970s. In contrast to this long tradition of combining games and simulations, it is usually not made explicit which kind of simulation is used, which models are fundament of the simulation, and which role the simulation and the models have in the game based scenario (e.g. non-player characters, environment, engines). This has led to a situation where research in modeling and simulation and development of games run in parallel – it is not clear how recent research results influence the interwoven development of both fields. This paper will start with an attempt to structure both fields, and to provide for a collection of defined terms, which help to relate modeling and simulation approaches to game development.

1 Introduction

The combination of computer games development and modeling and simulation has a comparably long tradition. The first journal „Simulation & Gaming“ has been available in March 1970. Interestingly, the first issues in this series combine research insights from the areas games, simulations, and teaching-/training systems – nowadays, this combination is often called „Serious Games“ (see `http://sag.sagepub.com/content/vol1/issue1/`). The projects and approaches described in the published articles in the early 1970s are usually influenced by psychological and social science research. This focus changed over the years from simulation technology toward the description of complex and graphically demanding computer games, probably because of advances in computer technology and available hardware. „Learning“ has not been the focus of investigation anymore.

Examples for the long tradition of combining games and simulations are business simulations [Big90] and flight simulators. The latter have not only been used in professional pilot training, but also have been available as pure computer games at the very early game consoles in the late 1970s (see `http://fshistory.simflight.com/fsh/index.htm`). Flight simulators are one example for systems which can address three aspects: they exist as pure simulations for experimental and research purposes (e.g. [Nor95]), they exist as serious games for training of behavior in new or critical situations and with new technology (e.g. [ZJL08]), and last but not least, they exist as pure games (e.g. `http://www.mobygames.com/game-group/flight-unlimited-series`). Sometimes, models from simulations can be re-used in serious games and in games (e.g. `http://www.`

microsoft.com/games/flightsimulatorx/) – however, the requirements for each system type differ.

The first level to structure the field of games and simulations is to have a look at the purpose of the developed software system. The purpose has a strong influence on the system design and the role of the simulation parts. Three directions can be distinguished in this context: education, insight, and entertainment. The second level refers to what exactly is simulated in the system. From the perspective of modeling and simulation, this level is related to the models used in the system. These can be physical models, models of the environment, or models of human or non-human beings. These models can be further divided into concrete kinds of models, i.e., continuous, discrete, or hybrid models. The third level of investigation is concerned with the simulation technique used. Herein, it is analyzed which types of simulation algorithms and protocols can be interesting for the realization of games. Approaches analyzed in this context are often not developed for games only, but for networking applications and distributed systems in general.

2 System Purpose

The purpose of the system at the border between games and simulation can either be education, insight, or entertainment. If the system is developed for educational purposes, the role of the simulation in the system is mainly to show and demonstrate. The simulation can be demonstrative, i.e. the user starts the simulation and can watch the simulation run, but he can not interfere. The simulation can also be realized in an interactive way. Here the span reaches from interaction with virtual environments or artificial characters (e.g. [RJ99]), "human in the loop" simulation, where the user can interact with the simulation, up to simple parameter changes (see e.g. [MDM08]). The user can be aware of the interaction – he knows about the embedded simulation techniques and is aware of the fact that he changes model parameters to influence the simulation run. Alternatively, the simulation is capsuled and hidden, e.g. behind complex graphical user interfaces. Then the user interacts with the interface and is not aware of the fact that he is interacting with a simulation system. The game-aspect in these systems is used intentionally to keep the user's motivation at a high level and to keep them engaged. These systems often come in the shape of game-based learning or edutainment systems. Users are in most cases children or young adults.

If the system is developed for research purposes, the role of the simulation in the system is to gain insight and knowledge. Systems at the border between games and simulations can also be found in this area. Here, the game-based part is in most cases the application scenario, which is used to test and apply a new development. One example are the Robocup games (see http://www.robocup.org/). Soccer playing robots, separated in different leagues, can be programmed in this context. Serious background of these applications are robot development, artificial intelligence, and programming. Moreover, games may also serve as application domain for simulation studies, e.g., when they are regarded as complex software systems with special characteristics that shall be investigated. For example, the network load which can be observed in multiplayer games like Half-Life or

Quake 3 can be empirically evaluated and modeled. Results of such studies can then be used for network simulations (e.g. [LBA04]). Similarly, the usability of simulation algorithms for a game genre can be investigated and evaluated by modeling and simulating the algorithm (e.g. in [CKFJ04]).

If the purpose of the system is entertainment, the gaming aspect becomes more important, whereas simulation is in the background. Criteria for game development differ from criteria for research or learning system development. Often, users (now as players) are not aware of the underlying simulation techniques. They interact with a virtual environment, with virtual characters, or with virtual devices. The players main interest are a good graphical representation and good performance of the game.

3 Models

In this section, it will be sketched which kinds of models can be used in games and simulation systems. The usage of the term model in everyday language is a bit blurred (see e.g. [Mar08]). Even in the field of “modeling and simulation”, there is some heterogeneity among the definitions of the term model. The perspective we take in our research is related to Minsky, who defined that a model M of a system S is a goal oriented picture of S [Cel91]. Generally, two types of models are distinguished: models that describe behavior over time, and models that should determine behavior. The first can be further distinguished in continuous, discrete, and hybrid models (see e.g. [ZPK00]), the latter are usually based on stochastic descriptions, e.g., Monte-Carlo methods.

The development of continuous models can be traced back to a time before the 17th century [GT99, p. 7]. After the first computers had been developed, the former analogue simulations have been displaced by continuous simulations with computer technique [rEM98]. Continuous models are formally described by differential equations. This implies that a continuous simulation theoretically simulates infinite state changes in each time interval. Continuous models are usually used in natural and engineering sciences, and can also be found in industry. In games, continuous models lend themselves to the description of physical processes, e.g., continuous descriptions can be used for weather models in flight simulators.

Discrete models describe the system functionality based on states and discrete state changes. First approaches in the direction of discrete modeling and simulation can be dated back to the 1940s [GT99, p. 7]. Well known discrete models in computer science are automata and state charts. Usually, application scenarios of discrete models are related to computer science research. As computers are discrete by nature, computer systems can comparably easy be characterized by discrete descriptions. Classical examples are network simulations for developing network protocols and components, and models for software testing. In most cases, games are not based on discrete models – even if the approach would lend itself to several game types. Examples would be business games, strategy games, and also simple games like Snake, Pacman, and Tetris. Using discrete models as a fundament of such games might facilitate game development and thus development cost.

Hybrid models combine continuous system descriptions with discrete state changes. These models can be used if the system of interest can mainly be described by a continuous formalism (a differential equation), but at a certain point in time, the state of the system changes toward another equation. These changes can, for example, be the crossing of a threshold, a time limit, or any other abrupt change. The hybrid modeling approach allows to describe system behavior in a comparably realistic way. In the context of games, this technique could be used in the development of Ego-Shooters, where the actors are discretized and can influence the environment. The latter might be modeled hybrid, as continuous states might be affected and changed by discrete interventions of the actors (e.g. movement of cars).

4 Simulations

Developers of games and simulation tools are facing several similar development challenges. Two main aspects can be distinguished, namely the plausible execution of models, i.e., the actual simulation, and the usage of efficient algorithms for common tasks (e.g. networking).

Comparing games and simulation systems with respect to the first aspect is difficult, as game developers are usually quite hesitant to publish newly developed algorithms (for economical reasons). Apart from that, analytic simulation strives to execute a model plausibly, in terms of some well-founded scientific semantics, whereas the execution of a model in a computer game merely has to *appear* plausible to the player(s). This motivates heuristics for physical game models instead of an actual physical simulation, which can save considerable computing efforts [YFR06]. Nevertheless, some game engines are also applied to modeling and simulation [WLG03], and modeling approaches may adopt approaches from gaming as well. For example, Banerjee et al. used the layered intelligence method from game programming to model crowds as multi-agent systems [BAK08].

The common ground of games and simulation systems is much clearer when considering the second aspect. For auxiliary tasks – such as data collection, distributed execution or load balancing – rather similar algorithms are applied in both fields, which makes the exchange of experiences particularly fruitful here. For instance, algorithms for Interest Management (IM), i.e., the management and provision of data in a distributed system, can be classified with respect to the characteristics of the application at hand. This application could be a distributed simulation or an online game. Another example are the online modes of Ego-Shooters and real-time strategy games. Usually, the former are implemented with network protocols like UDP (User Datagram Protocol), which allow for high speed connections but are not reliable. The latter are mainly based on the reliable but slower TCP (Transmission Control Protocol). As a result, both genres exhibit different IM semantics, which can be addressed by different adaptive IM methods from the field of parallel-distributed discrete-event simulations (PDES) [MT07]. The Torque Network Library [Gar] reflects the similarity of networking problems by supporting both parallel-distributed simulations and online-modes of games. A further point of contact is the field of synchronization. For PDES as well as for games the *Local Causality Constraint* (LCC), i.e., the correct execution order of events has to hold [Fuj00, CKFJ04]. In multiplayer games, not only the synchronization plays a major role, but also aspects like cheating, which could be handled by well-known PDES methods like replication [CCRJ06, WST08].

5 Summary

"Simulation" is a commonly used keyword. Diverse groups of scientists and developers in research and industry make use of simulation technology. Thereby, simulation has a different notion in the diverse fields, which hampers the exchange of knowledge . Firstly, we introduced important "names" from a basic modeling and simulation research perspective, and, secondly, we used this point of view for presenting some thoughts on "game" developments. Hopefully, this helps to support the various groups involved, so that they obtain an easier access to the knowledge of the others.

Literatur

[BAK08] Bikramjit Banerjee, Ahmed Abukmail und Landon Kraemer. Advancing the Layered Approach to Agent-Based Crowd Simulation. In *22nd WS on Principles of Advanced and Distributed Simulation*, Seiten 185–192, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[Big90] W.D. Biggs. *Guide to Business Gaming and Experiential Learning*, Kapitel Introduction to Computerized Business Management Simulations, Seiten 23–35. 1990.

[CCRJ06] Fabio R. Cecin, Claudio, Solon Rabello und Jorge. A peer-to-peer simulation technique for instanced massively multiplayer games. In *DS-RT'06. Tenth IEEE International Symposium on Distributed Simulation and Real Time Applications*, Seiten 43–50, 2006.

[Cel91] Francois E. Cellier. *Continuous System Modeling*. Springer-Verlag, 1991.

[CKFJ04] Eric Cronin, Anthony R. Kurc, Burton Filstrup und Sugih Jamin. An Efficient Synchronization Mechanism for Mirrored Game Architectures. *Multimedia Tools and Applications*, 23(1):7–30, May 2004.

[Fuj00] Richard M. Fujimoto. *Parallel and Distributed Simulation Systems*. John Wiley and Sons, 2000.

[Gar] Garagegames. *Torque Network Library Reference*. http://opentnl.sourceforge.net/doxytree/.

[GT99] Nigel Gilbert und Klaus G. Troitzsch. *Simulation for the Social Scientist*. Open University Press, 1999.

[LBA04] Tanja Lang, Philip Branch und Grenville Armitage. A synthetic traffic model for Quake3. In *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, Seiten 233–238, New York, NY, USA, 2004. ACM.

[Mar08] A. Martens. Simulation in Teaching and Training. In *Encyclopedia of Information Technology Curriculum Integration*. IGI Global, 2008.

[MDM08] A. Martens, H. Diener und S. Malo. Game-based Learning with Computers – Learning, Simulations, and Games. *Transactions on Edutainment*, to appear 2008.

[MT07] Rob Minson und Georgios Theodoropoulos. Adaptive Support of Range Queries via Push-Pull Algorithms. In *PADS '07: Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation*, Seiten 53–60, Washington, DC, USA, 2007. IEEE Computer Society.

[Nor95] Ken A. Norlin. Flight Simulation Software at NASA Dryden Flight Research Center. NASA Technical Memorandum 104315, NASA, 1995.

[rEM98] Karl Johann Åström, Hilding Elmqvist und Sven Erik Mattsson. Evolution of continous-time modeling and simulation. In *Proceedings of the 12th European Simulation Multiconference*, Seiten 9–18, Manchester, UK, Juni 1998.

[RJ99] J. Rickel und W. L. Johnson. Animated Agents for Procedural Training in Virtual Reality. *Applied Artificial Intelligence Journal*, Seiten 343–382, 1999.

[WLG03] Jijun Wang, M. Lewis und J. Gennari. A game engine based simulation of the NIST urban search and rescue arenas. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, Jgg. 1, Seiten 1039–1045 Vol.1, 2003.

[WST08] Steven Daniel Webb, Sieteng Soh und Jerry Trahan. Secure Referee Selection for Fair and Responsive Peer-to-Peer Gaming. In *2nd Workshop on Principles of Advanced and Distributed Simulation (PADS 2008)*, Seiten 63–71, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[YFR06] Thomas Y. Yeh, Petros Faloutsos und Glenn Reinman. Enabling real-time physics simulation in future interactive entertainment. In *sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, Seiten 71–81, New York, NY, USA, 2006. ACM.

[ZJL08] L. Zhang, H.Z. Jiang und H.R. Li. PC-Based Human-In-the-Loop Simulation for Flight. *Applied Mechanics and Materials*, 10–12:476–482, 2008.

[ZPK00] Bernard P. Zeigler, Herbert. Praehofer und Tag G. Kim. *Theory of Modeling and Simulation*. Academic Press, London, 2000.