M3 Project - Actor Based Simulation in Virtual Worlds

Helge Rosé, Torsten Asselmayer-Maluga, Peter Frank, Andreas Hoheisel and Bertram Walter

Fraunhofer Institute for Computer Architecture and Software Technology
Kekuléstr. 7, D-12489 Berlin, Germany, Helge.Rose@first.fraunhofer.de

Abstract: M3 Simulation is a simulation approach that actively involves real human actors in a virtual reality environment based on multidisciplinary simulation models. This actor based concept aims at integrated simulation of complex processes belonging to heterogeneous levels of reality, paying specific attention to real human behavior. The entities of the real system under consideration are represented in a consistent object-oriented way by agents simulated by environmental models. This distributed network-capable implementation of the M3 system combines environmental modeling with the intuitive graphic representation, complex communication and interaction structures of virtual worlds in order to support the process of investigation of complex real world systems.

Keywords: actor based simulation, interactive simulation, environmental models, virtual worlds, sustainable development, http://mmm.first.fraunhofer.de

1 Introduction

Simulations of real world systems are currently developing to become a decisive instrument for IT-supported problem solving in science, economy and society. The main objective of the M3 project is the realization of a distributed, interactive real world simulation system involving a large number of real human actors in a virtual reality environment implemented by environmental simulation models (Rosé and Jugel [2001]).

A real world system is characterized by an interplay of two domains: a natural environment and a socio-economic system. Simulation is faced with a particular set of problems for each domain. This twofold character of problems has to be reflected by the design of a simulation system.

In case of the environmental domain there exists a large manifold of validated simulation models describing distinct aspects of the environment. Jorgensen et al. [1994] gives a selection of over 400 scientifically-founded models from a set of several thousands reflecting all kinds of environmental subsystems.

On the one hand side, the validated environmental models are restricted to a specific region and purpose. Each of them needs a particular preparation of input data from measurements. This forces the requirement of data support by local measurement providers. Most of the sophisticated models are monolithic implementations with rigid interfaces which necessitate a large effort to apply such a model to a new region. Even when all input data are available, the run of such a model may fail because often a very specific 'fine tuning' work has to be done.

On the other hand, there exists a large set of regional validated models ready for application which are implemented and supported by the local research institutes. But this large set of models is not open to public access and is not connected to a network of local simulation sites.

These problems of environmental simulation formulate the first guiding rule for the structure of the simulation system: The structure should be *component-based, distributed and network-capable*. A network of local simulation components makes it feasible to utilize the regional available knowledge and simulation models. The regional institutes and research groups can integrate, extend and support their already used models within the framework of simulation network without loosing control or competence.

In case of the socio-economic domain, simulation is approaching to become a widely used tool and introduces new ways of investigation based on ideas about emergence of complex behavior from relatively simple activities. There exists a broad spectrum of approaches reaching from deterministic system dynamics and stochastic models to multi-level modeling and multi-agent systems (Nigel and Troitzsch [1999]).

The description of socio-economic systems by these approaches has the advantage of a controllable and reproducible simulation of integral behavior neglecting individual fluctuations. This averaging description is appropriate at stages of the system evolution near the equilibrium, when small fluctuations can not change the overall behavior. Especially at times of change, when the whole system becomes sensitive to individual actions, this approach is limited because a computer simulation reflects programmed properties only. But real people can make unexpected actions which become important at bifurcation points of socio-economic development (Haken [1996]).

This formulates the second decision for the structure of the proposed simulation system: The socio economic behavior should not be simulated but realized by involving a large number of real people - actors – in a Virtual Reality (VR) environment implemented by multidisciplinary simulation models Every actor has a representant in the VR environment which lives and works, producing pollution and getting diseases.

A well-known example for this are the Multi User Dungeons (MUDs) (Bartle [1990]) that are spread on the Internet, where hundreds of players walk through a world with thousands of rooms described by text, communicate and interact with each other in order to solve mysteries that are posed in the various rooms. Multiple communication and interaction structures of actors – despite they are free from any graphic representation – are what render MUDs virtual representations of social systems.

Currently, there is a growing interest to use this idea of interactive multi-player games in scientific applications. Interesting applications are multi-player business games (Hardman et al. [1995]) and internet forum for participatory group learning and model improvement (Hare et al. [2001]). The business game is restricted to interactions between human and simulated players only, whereas the internet forum uses inter-player facility to establish a stakeholder discussion process.

Following this line of interactive approaches, the focus of the M3 project is the development of software techniques which support the investigation of the interplay between environment and socio-economic system. To facilitate this, one has to implement a software infrastructure which is able to maintain a network of environmental models establishing a virtual image of the natural environment which can be interactively affected by a large number of participating persons. These *actors* actively participate in the simulation by their actions and can develop strategies securing long-term development. On the other hand, they can experience possible consequences of their behavior and, thus, improve their strategic decisions in a process of evolutionary learning. Recently it was shown, that multi-player online games, which are able to attract a number of several thousands of players, can be studied like any real economy systems, even though these games are fantasy worlds. This is because of the social importance attached to the game by its players (Castronova [2001]). To reach such large number of motivated participants, the system has to realize a intuitive, graphical representation of the simulation data by means of Virtual Reality techniques.

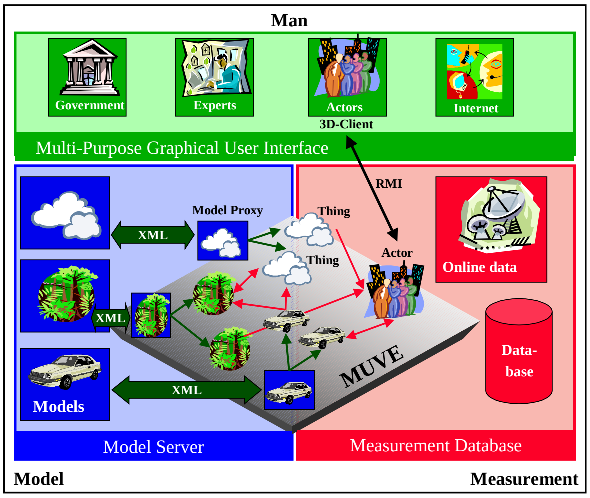

Figure 1. Structure of the M3 system.

A simulation using a network of simulation *models* driven by up-to-date *measurement* data of real system under consideration and involves *man* as acting and experiencing part we refer to as **M**an-**M**odel-**M**easurement simulation (Rosé [8-11 January 2001]). The M3 system (see Fig. 1) is divided into four parts: the Multi User Virtual Environment (MUVE), the model server, database and the user interfaces.

The paper is structured as follows. In the next section we describe the Multi User Virtual Environment (MUVE) which is the heart of the system. Every object has a representant in the MUVE. Then we describe the structure of the model server including the wrapper concept to control the simulation models. A short description of the database server follows. The two last sections explain the structure of the Multi-purpose Graphical User Interface and the method of development the M3 system.

2 Multi User Virtual Environment

The MUVE is the logical representation of the real world environment under consideration by a *network of Things* (Figure 1).

The Things are communicating software components which simulate the real world objects and their interactions. A crucial advantage of modeling a real world system in this way is given by the possibility to map the real world of objects into the MUVE in a consistent *object-oriented* manner. Each Thing can be characterized by its current *state*, possible *processes* of changing this state and its *relations* to other Things describing the interactions of the real world objects.

With reference to the autonomous behavior of the Things the MUVE may be classified as a Multi Agent System. Even though the Things are very simple agents without a goal-based behavior, the object-oriented structure of the MUVE enables an easy integration of any kind of complex agents.

At the current stage of the project a basic hierarchy of Thing classes has been implemented in Java 2.

```
Thing
RelatedThing
ChangeableThing (implements Changeable)
LocatedThing (implements Viewable)
DynamicalThing (implements Changeable)
MoveableThing (implements Moveable)
MechanicalThing
Actor
```

The class `RelatedThing` extends `Thing` by the ability of communication to others via relations. The relations are realized by Delegation Based Event Handling (Listener Design Pattern). All other basic classes inherit this ability. `ChangeableThing` implements the interface `Changeable` providing the method `changeState`. This method determines the processes of the Thing. A `LocatedThing` has a place and orientation. Additionally it implements the interface `Viewable` which associates a graphical representation (image, 3D-mesh, animation) with the Thing. The classes `DynamicalThing`, `MoveableThing` and `MechanicalThing` formalize the different levels of change of a located Thing. `MoveableThing`s are able to sense collisions and `MechanicalThing`s behave like Newtonian rigid bodies. Concrete classes which extend the basic ones are for example `AgrarianField`, `Building`, `Car`, `Vegetable` and `WeatherStation`.

The class `Actor` represents the participating real persons. It provides mechanical and metabolic properties like energy consumption, nutrition content and health index. Additionally it implements all necessary networking stuff for connection with the 3D-client running at the users home PC. At the time the M3 systems simulates a 2x2 km range of the Berlin Alexanderplatz surrounded by an agricultural area. This test version has some characteristical buildings, streets and agent-controlled cars producing pollution.

The external environmental models are integrated into the MUVE by specialized Things. These *model-proxies* are dynamically instantiated and connected to the corresponding state variables of the Things via relations. Every external model has its own model-proxy which transmits the simulation data from model to the state variables of its related Things.

By this concept, the models are connected via a communication structure induced by the relations of the corresponding Things. The control and data communication between the model and its model-proxy is realized by a specialized markup language which employ the XML technology for easy extension and validation (Jugel [8-11 January 2001]).

New or alternative external models can easily be integrated by disconnecting the current model-proxy and connecting the new one to the respective variables of the Things. This enables a fast evaluation and test of alternative model implementations without the necessity to stop the whole system and recoding of the Things.

3 Model server

In the M3 system a model server is used to provide a distributed network of environmental models connected via generic control and data mapping interfaces. A general problem that is addressed here, is the integration of heterogeneous implementations. Environmental models are written in different programming languages like C or FORTRAN and use different data formats. Most of the models are normally used as stand-alone programs on a specific platform. The models are generally not designed to be used in a coupled simulation in conjunction with other models that may use another programming language or even require a special operating system. To integrate available environmental models we have to define the data exchange and control protocols.

The M3 system uses the concept of model wrapper written in Java. The wrapper encapsulates the external simulation program and defines a standardized input/output interface. We decide to use XML (Extensible Markup Language) as communication language between the simulation program and the wrapper. In most cases a tight coupling scheme is used in environmental simulation, and shared memory is used for the communication between the coupled models. Tight coupling generally requires a lot of effort when integrating the models, and often it is necessary to implement all models in the same programming language. In the M3 simulation we use a loose coupling scheme instead, with asynchronous communication between the coupled models. The model coupling is network-oriented and based on semistructured or unstructured data exchange formats (e.g. XML). This kind of model coupling is very flexible and makes it easy to reuse single models in another context.

In the M3 simulation we use dynamic coupling with late time binding during runtime, so that the cou-

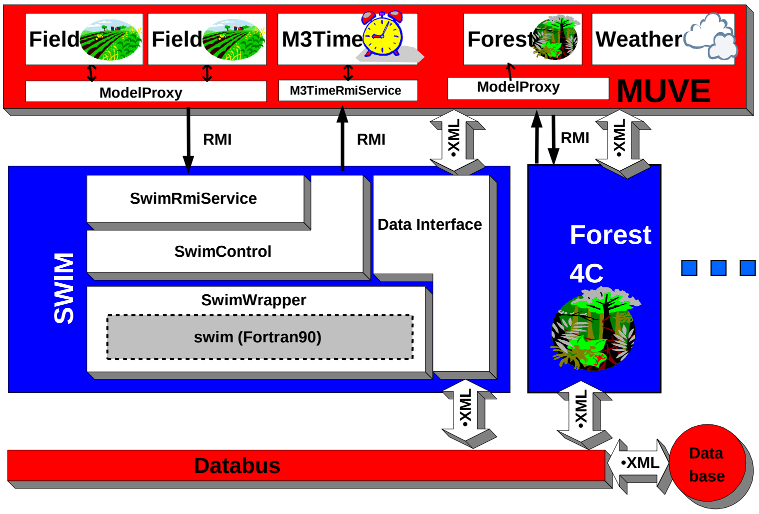

Figure 2. Model server of the M3 system is an extensible network of environmental simulation models communicating by XML messages.

pling does not have to be defined when compiling. This enables us to select or exchange simulation models during runtime without the necessity of restarting the simulation or recompiling the simulation program. The models of the whole simulation are distributed over several concurrent processes that use online communication for bidirectional or unidirectional coupling.

The control of the simulation by the model server in the M3 system is realized via a RMI (Remote Method Invocation). The wrapper of the corresponding model provides a RMI service which can be accessed via the network to start, control and stop the execution. The Java wrapper uses standard input (stdin) and standard output (stdout) of the simulation program as interfaces for the communication with legacy code that is not written in Java. A substantial advantage of this type of interface is that almost every programming language implements the possibility to write to stdout and to read from stdin. Single model components that use stdout/stdin for their controlling and data transfer can very easily be tested and validated by means of scripts or by simply invoking method calls at the command line. After the start of the simulation via the RMI service, the program accepts method calls and data in the XML format. Currently the M3 system includes the following models:

Swim (Soil and Water Integrated Model) integrates hydrology, vegetation (e.g. crop growth), erosion, and nutrient dynamics at the watershed scale. It was developed by the Potsdam Institute for Climate Impact Research (Arnold et al. [1993], Krysanova et al. [1989]). Implementation: FORTRAN 77/90.

4C a model to simulate the growth of trees in the forest including hydrology. It was also developed by the Potsdam Institute for Climate Impact Research (Bugmann et al. [1997]). Implementation: FORTRAN 90.

Wgen the weather generator developed by Richardson and Wright. On the basis of some statistical data, it produces daily weather by a random process (Richardson and Wright [1984]). Implementation: C.

REWIMET a hydrostatic, three layer model for the propagation of gases in the atmosphere including the current meteorological situation. The model is coupled with a lagrangian model to simulate the propagation of PM_{10} particles (Mieth et al. [1998], Unger et al. [1998]). Implementation: C.

Impact an impact model to simulate the impact of the environment on the health of the people due to pollution of air and food (Beger et al. [2001]). Implementation: Java.

Traffic a mesoscopic traffic simulation model (Schmidt et al. [1998]). Implementation: Java.

4 Database server

The database server contains all information needed to start the models as well as configurations of the Things and the MUVE. By the same time, it also logs all relevant informations generated by the system during a simulation run. The network-based

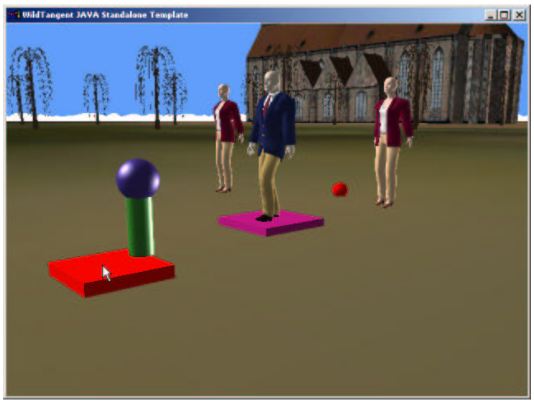

Figure 3. Graphical representation of the world. The 3D-Client is running on the home PC of the participant. The client is connected to the MUVE via a low bandwidth internet connection.

database service is provided by a MySQL server that can be accessed by the MUVE using a JDBC driver. The data exchange between database, the model wrapper and the MUVE is implemented by standard SQL calls. The data can be retrieved from the database server for external use by a standard web browser.

5 Multi-purpose Graphical User Interface

The MUVE and the model server generates a simulated environment, which is populated by actors. To get in contact with the M3 simulation, the user has to use a special adapted configuration of the Multi-purpose Graphical User Interface visualizing the simulated environment in a graphical form. In dependence of the current user (expert, actor, politician, etc.), one gets different *views* on the Virtual World showing diagrams or a 3D representation (see Fig. 3). A future task will be the implementation of suitable expert GUIs which can easily realized by available XML tools using the direct XML output of the models.

For the visualization the Wildtangent 3D-engine is used. It offers 3D acceleration using the DirectX interface in Microsoft Windows. Additionally it provides high level methods for building virtual worlds and for the interaction in them. For example the animation of the actors is done with 3D Studio Max (predefined movements) and exported to Wildtangent and there replayed with a single command. The polygonal and textural structure of the actors is modeled in Poser, whereas non movable objects (trees, houses) are incorporated into the environment using the Wildtangent converter for the very popular 3DS object format. Because Wildtangent offers a Java interface to its rendering engine, we could reuse the RMI classes for the network communication between the client and the MUVE.

To provide a uniform environment for all players the position of all movable objects (e.g. actors, cars) is controlled through the MUVE. It calculates the new positions according to collisions and environmental influences (e.g. friction, gravity) and sends these data to the clients, which will then update the position of the visible objects. Because of this structure the controlling of the actor through the user is done in an indirect way. Instead of specifying movement directions the player only sets movement points to which the actor is walking on its own. This way inconsistencies between player positions in the client and the MUVE can be minimized and the player can shift his attention from controlling his actor to interacting with the virtual world.

6 Conclusions

The realization of the M3 project is an encouraging and very challenging enterprise. Doubtlessly, the effort necessary for design and implementation of a M3 System can not be performed by separated groups of software engineers, environmental scientists or modelers. Successful realization necessitates a far-reaching cooperation of all involved scientific fields. In fact, the interactive character of the M3 world itself provides an efficient environment for development, fast communication and cooperation. Because the M3 Framework is not only a "software" but a simulated world with real people, it has the ability to *evolve itself*. This kind of software development has proven its power by the success of open software projects like Linux and GNU (Raymond [2001]).

All users are able to contribute to the system by adding new implementations. The development process is realized by running several test instances of the whole system and the evaluation with respect to functionality, stability and utility. The best modifications flow into the proper instance of the system. This way of development is in fact a process of *evolutionary improvement*. A recent two-year empirical study at the Harvard Business School has shown, that the best way of software development is this evolutionary one (MacCormack [2001]).

Currently a prototype of the M3 system with limited functionality has been implemented showing the technical feasibility of the M3 simulation and now one has to focus on substantial questions. The experience with multi-player online games, like EverQuest, shows that such systems are able to reflect some aspects of realistic behavior, but it is still to be investigated if also complex socio-economic systems can be described by this approach. A further difficulty is the coupling problem of different environmental models that use different assumptions, techniques and boundary conditions. Thus the M3 system can be applied for investigating a limited class of problems and has to be tailored according to the requirements of the interacting whole system.

Acknowledgements

We would like to thank Prof. Hans-Joachim Schellnhuber and colleagues at the Potsdam Institute for Climate Impact Research for fruitful cooperation and our colleagues Birgit Kwella, Mathias Schmidt and Steffen Unger for their enthusiasm and engagement in trying to make M3 becomes reality. This work is supported by the Helmholtz Society and the German Federal Ministry of Education and Research, (FKZ: 01SF99147).

References

J.G. Arnold, P.M. Allen, and G. Bernhardt. A comprehensive surface-groundwater flow model. *Journal of Hydrology*, 142:47–69, 1993.

R. Bartle. Interactive multi-user computer games. *British Telecom Report*, 1990.

E. Beger, D. Bieninda, and S. Gester. Modul Impactmodelle im M3-System des Projektes GLOBALSIM. *Technical report, IBB Ingenieurbüro Beger*, 2001.

H. Bugmann, R. Grote, P. Lasch, M. Lindner, and F. Suckow. A new forest gap model to study the effects of environmental change on forest structure and functioning. In G.M.J. Mohren and K. Kramer, editors, *Global Change Impacts on Tree Physiology and Forest Ecosystems*, pages 255–261. KLuwer Academic Publishers, 1997.

E. Castronova. Virtual worlds: A first-hand account of market and society on the cyberian frontier. *CESifo working paper No. 618, subm. to netnomics*, 2001.

H. Haken. Synergetik und Sozialwissenschaften. *Ethik und Sozialwissenschaften. Streitform für Erwägungskultur*, 7:587–594, 1996.

L. Hardman, G. van Rossum, and A. van Bolhuis. An interactive multimedia business game. *Journal of Intelligent Systems*, 5:151–177, 1995.

M. Hare, N. Gilbert, D. Medugno, T. Asakawa, J. Heeb, and C. Pahl-Wostl. The development of an internet forum for long-term participatory group learning about problems and solutions to sustainable urban water supply management. In L.M. Hilty and P.W. Gilgen, editors, *Sustainability in the Information Society*, pages 743–750, Marburg, 2001. Metropolis Verlag.

S. E. Jorgensen, B. Halling-Sorensen, and S. N. Nielsen. *Handbook of Environmental and Ecological Modeling*. Lewis Publishers, Boca Raton, London, New York, Tokyo, 1994.

M. L. Jugel. Enhancing MUVEs: Connecting virtual objects with environmental simulation. In K. Bellmann and C. Landauer, editors, *Proceedings of the International Conference on Virtual Worlds and Simulation*, Phoenix, Arizona, 8-11 January 2001.

V. Krysanova, J. Roosaare A. Meiner, and A. Vasilyev. Simulation modeling of the coastal waters pollution from agricultural watershed. *Ecological Modeling*, 49:7–29, 1989.

A. MacCormack. Product-development practices that work: How internet companies build software. *MIT Sloan Management Review*, 42:75–84, 2001.

P. Mieth, S. Unger, and M.L. Jugel. An environment simulation and monitoring system for urban areas. *TRANSACTIONS of SCS*, 15:115–121, 1998.

G. Nigel and K. G. Troitzsch. *Simulation for the Social Scientist*. Open University Press, London, 1999.

E. S. Raymond. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. OReilly & Associates, 2001.

C.W. Richardson and D.A. Wright. Wgen: A model for generating daily weather variables. *Technical report, U. S. Department of Agriculture, Agricultural Research Service*, 1984.

H. Rosé. M3-Simulation - Multidimensional Modeling of Sustainability Strategies in Virtual Worlds. In K. Bellmann and C. Landauer, editors, *Proceedings of the International Conference on Virtual Worlds and Simulation*, Phoenix, Arizona, 8-11 January 2001.

H. Rosé and M. L. Jugel. The M3-Project - Living in Simulated Worlds. *ERCIM News*, 45:38–39, 2001.

M. Schmidt, R.-P. Schäfer, and K. Nökel. SIMTRAP: Simulation of traffic-induced air pollution. *TRANSACTIONS of SCS*, 15:122–132, 1998.

S. Unger, I. Gerharz, P. Mieth, and S. Wottrich. HITERM- high-performance computing for technological risk management. *TRANSACTIONS of SCS*, 15:109–114, 1998.