The M3-Project: Distributed Environmental Simulation and Virtual Reality

T. Asselmeyer-Maluga*, P. Frank, A. Hoheisel,
H. Rosé and B. Walter
GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany

April 27, 2001

Abstract

Environmental simulation models contain mostly a lot of single models coupled to each other to reflect the complexity of our World. But an important factor is excluded in such simulations tools: the unpredictable human factor. In this paper we will describe a new simulation tool: the **M3 simulation (man machine measurement).** This simulation includes real people into the simulation by a Virtual Reality framework.

Papertype: FULL Paper, Session: Environmental Science

1 Introduction

Large simulation models are used for environmental modeling, in particular for pollution analysis and forecast in the air domain. Many of them already run in parallel on parallel computers or clusters of workstations or PC's. The term 'distributed' is often used in this context to name a parallel simulation. On the other hand, pre- and post-processing, i.e. preparation of different kinds of input data and visualization and interpretation of the results are often an even more challenging task. The support of different institutions

*email:torsten@first.gmd.de, http://m3.first.gmd.de

like local authorities, weather service, public information sources is required. Most of the sophisticated models nowadays require very special input data, run locally with strongly defined interfaces only and a large amount of work is necessary to apply such a model to a new domain or link it with different input generators or models. Even then 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, for many situations already exist tuned models, ready for application, but not open to public access and not ready to work with a given input data set. We outline here a system, which may allow in future to run the most appropriate model for a given task. A distributed simulation with respect to this means, that input data, models, post-processing and visualization tools in general are distributed over different institutions, universities etc. The necessary work flow, data transfer and wrapping is maintained by the system, the exchange of data will be done via the Internet by means of special WEB protocols, which also will provide access to computing power in computing centers.

Another open problem in Environmental simulation is the inclusion of the socio-economic component in the simulation. Most people which deal with this problem try to use agents reflecting the properties of people. But this approach has some disadvantages because the agents reflect only the programmed properties. In contrast to this, real people can make unexpected actions which can never be simulated. For that purpose, we decide to include real people in our simulation framework by use of the Virtual Reality framework. We call such simulation a M3 simulation [1]. The technology is borrowed from the Multi User Dungeons (MUD's) [2] but now with a 3D graphical output. Every user has a representant in the Virtual Reality (VR) which lives, works and relax in this World producing pollution and getting diseases. These complex interactions between people and environment creates the non-linearity in our model necessary for every environmental simulation model. A simulation system that considers our position within the development should therefore:

- Mirror the **active role of people** in the process by contributing answers to the questions: Which solutions and strategies will be invented by actors in a World that responds with far-reaching alterations of the natural environment to their economical and social activities?

- It has to have a **educational function**, rendering the changes and personal consequences intuitively understandable.

Of course we remark also to get scientific knowledge in nearly all fields (economy, natural science, humanities, etc.) by usage of the M3 system.

The M3 system (see Fig.1) [1, 2] is divided into three parts: the MultiUser Virtual Environment (MUVE), the model server and the database. To reflect the universality of our system, we use as the superior program language **JAVA** including the extensive program libraries. The control of the simulation programs is handled by the JAVA library **RMI** (Remote Method Invocation) whereas a simple **GUI**(Graphical User Interface) can be created with the **SWING** library. For the data base, we use the **mySQL** server with the JAVA library **JDBC** (JAVA DataBase Connectivity) to control the exchange of data.

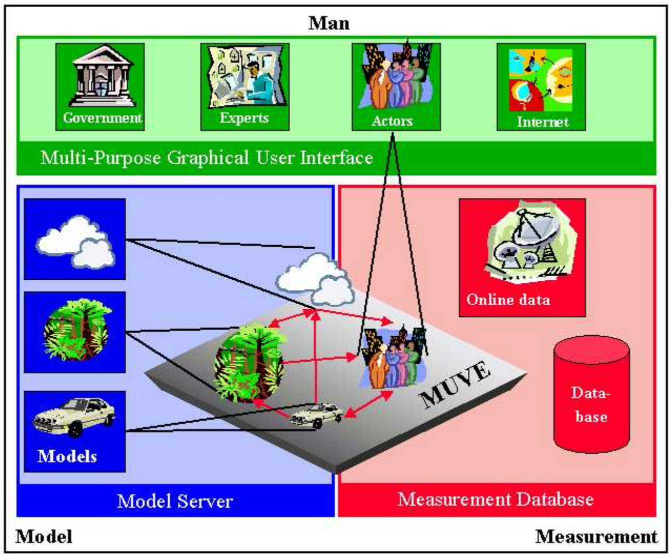

Figure 1: The structure of the M3 system

The paper is structured as follows. In the next section we describe the MultiUser Virtual Environment (MUVE) which is the heart of the system. Every object has a representant in the MUVE whereas the actors are so-called arvatars. 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. In the last section we explain the structure of the Multi-purpose Graphical User Interface to view as an actor the state of the Virtual World in an intuitive way.

2 MultiUser Virtual Environment (MUVE)

The MUVE is the heart of our simulation system. It represents the logical structure of the simulated World and coordinates the communication between the following components:

- **The Multi-Purpose Graphical User Interface**: realizing the visualization of the simulated World for different user groups (actors, experts, etc.).
- **The Model server**: controlling the connection between the distributed simulation models.
- **The (Measurement/Parameter) Database**: including all relevant parameter to start and control the simulation models as well as the logging of all relevant data and parameter of the Virtual World.

The MUVE is the logical realization of the mapping between the real World and the simulated World in a consistent object-oriented manner. That means, every considered object has a representation in the MUVE reflecting all properties of the object in the real World. Then, we collect all properties of an object into its state. The changing of this state and its relation to the states of all other objects defines the interaction in our system. By this mechanism we get automatically the coupling between our simulation models induced by the coupling of the objects in the MUVE. An additional model which should be integrated into the framework has to respect the connection structure of the already established object classes of the MUVE. The problem, to include a new model in a compatible way, can be solved by a class hierarchy of MUVE-objects. Mostly, there is the problem to restart the (static) program after the changing of the World. To create a more flexible MUVE, we use a system that allowed dynamic creation and destruction of all Virtual objects while keeping the World on-line.

3 Model Wrapper and Model Server

Simulation models are mostly written in different languages like C, FORTRAN etc. and use different formats for the data exchange. To standardize the data exchange and the control of the simulation model we use the concept of a model wrapper written in JAVA. The wrapper encapsulate the simulation to control the simulation program by sending commands and getting a standardized input/output. We decide to use XML (Extensible Markup Language) as communication language between the simulation program and the wrapper.

Example of XML output (Weather generator Wgen):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE wgen SYSTEM "http://m3.first.gmd.de/m3/model/wgen/dtd/wgen.dtd">
<!-- data output from wgen -->
<wgen>
 <version>1.1_m3</version>
 <methodResponse nr="1" name="step">
  <dataIO year="1984" day="1"/>
 </methodResponse>
 <methodResponse nr="2" name="getWeather">
  <dataIO year="1984" day="1">
   <coordinate  x="134544.54" y="245454.35" z="150.00">
    <weather>
     <precipitation double="0.0090000"/>
     <solarRadiation double="12000000.0"/>
     <dayLength double="36295.0"/>
     <temperatureMaximum double="280.0000000"/>
     <temperatureAverage double="279.0000000"/>
     <temperatureMinimum double="278.0000000"/>
    </weather>
   </coordinate>
   <coordinate  x="154544.54" y="245354.35" z="130.00">
    <weather>
     <precipitation double="0.0070000"/>
     <solarRadiation double="11000000.0"/>
     <dayLength double="46235.0"/>
     <temperatureMaximum double="280.0000000"/>
     <temperatureAverage double="279.0000000"/>
     <temperatureMinimum double="278.0000000"/>
    </weather>
   </coordinate>
  </dataIO>
 </methodResponse>
</wgen>
```

The usage of JAVA as a superior computer language enforces us to include the extended class library to realize a distributed systems. The control of the simulation by the model server (see Fig. 2) in the M3-system is realized via a RMI (Remote Method Invocation) server with XML syntax coded method calls. The JAVA wrapper of the corresponding model provides via a fixed port the RMI service which can be accessed via the Internet to start, to control and to finish the simulation. After the start of the simulation via the RMI service, the program accept method calls and data in the XML format.

A central element in controlling the different models is the central model

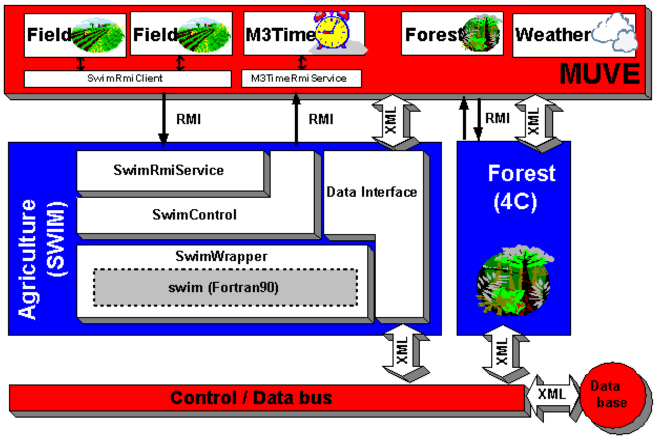

Figure 2: Model server with Model wrapper

time also called M3-time. This time indicates the current time of the Virtual World and controls by a clock pulse the synchronized coupling between the models. In the beginning of every day, the M3 time sends a broadcast pulse to every model to run the model for the updated time interval.

Currently we include the following models:

- *Swim* - an agriculture model including the hydrology of the area. It was developed by the PIK (Potsdam Institute for Climate Impact Research) [3, 4].
- *4C* - a model to simulate the forest also including the hydrology. It was also developed by the PIK [5].
- *Wgen* - the weather generator developed by Richardson and Wright. On the basis of some statistical data, it produces daily weather by a random process [6].
- *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 [7, 8].

- *Impact* - an impact model to simulate the current impact on the people from the environment [9].

- *Mobility* - a traffic simulation model to get the information about the traffic in the model [10].

Every model is controlled via a wrapper described above.

The model server receives the information from the MUVE about the exchange of data between models and the direct coupling of models. Then, this server sends some corresponding information to the model wrapper to realize this coupling. Thus after initialization, the models exchange data and control pulse directly. Then the model server is only used for the coupling between the objects in the MUVE and the simulation models.

4 Database (Measurement) server

The database server is one of the main ingredients for the start of the "World". It contains information about the initialization of the models as well as the parameters and other useful data. By the same time, it also log all relevant informations like parameter and data about the Virtual "World" (see Fig. 3). The database server is realized by a MySQL server with the

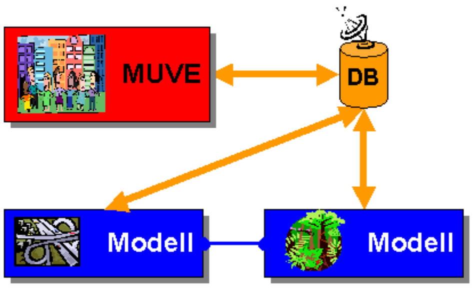

Figure 3: couplings of the data base

JDBC environment for the coupling to JAVA. This enforces us to write some general routines for the data exchange between database and models on the basis of the SQL language. The database server has to be started at first to provide the initialization of the other services.

5 Multi-purpose Graphical User Interface

By the methods described above, MUVE and the model server generates a simulated World, which are occupied by real actors represented by so-called arvatars in the Virtual World. To get in contact with the M3 World, the user has to use special adapted configuration of the Multi-purpose Graphical User Interface visualizing the simulated World. In dependence of the current user (expert, actor, politician, etc.), one gets different **views** on the Virtual World showing diagrams or 3D pictures (see Fig. 4). The model output

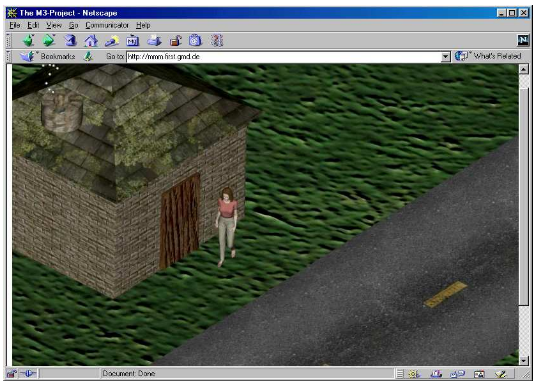

Figure 4: GUI with 3D representation as seen by an actor

in the XML format can be directly used by some visualization tools. The evaluation of the representation for the Virtual World viewed by an actor should be fulfill some contradicting aims:

- The representation should be objective and close to reality. Thus, the graphics must be detailed enough.
- The visualization tools should be computer- and system-independent. That enforces us to use JAVA and/or VRML.
- The graphical client should be react without nearly any time shift to allow complex changing of objects. Thus, the data have to be small enough to use a modem.

Furthermore, the graphics has to be changed fast enough to catch the interest of the user. Thus, the graphic acceleration of 3D cards must be used to fulfill these requirements. We have to use also the compression of texture (e.g. S3TC, FXT1) and geometry data.

The represented objects divide into two central groups:

- movable objects (people(actors), cars, trains, etc.) and
- unmovable objects (buildings, fields, forest, streets, lights, etc.)

In these two groups, we have to distinguish between animated (e.g. lights, walking people) and non-animated objects.

We choose the Wildtangent Kid as 3D Engine because it works together with Windows, supports the 3D acceleration with DirectX, has a JAVA interface and runs as an applet in the web browser. For the simulation of the arvatars, we use the Poser software to create for instance the movement of people. Unmovable objects are created by the 3DStudioMax program.

6 Conclusion

The M3 project is the trial to include the unpredictable human factor into an environmental simulation. It is used to simulate sustainable development strategies to create for every people an educational view on global problems in our World. It has the great advantage to use and include standard environmental simulation models in a distributed system. For further information about the M3 project and the current state please contact our web page http://m3.first.gmd.de.

7 Acknowledgements

We would like to thank Prof. A. Sydow and M.J. Jugle for the fruitful discussions. Many thanks to our colleagues B. Kwella, M. Schmidt and S. Unger for their engagement to realize the M3 project.

References

[1] H. Rose. M3-simulation: Multidisciplinary modeling of sustainability strategies in Virtual World. In *Proceedings of the International Conference on Virtual Worlds and Simulation*, Phoenix, January 2000.

[2] M. Jugel. Enhacing MUVEs: Connecting Virtual objects with environmental simulation. In *Proceedings of the International Conference on Virtual Worlds and Simulation*, Phoenix, January 2000.

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

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

[5] 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*, chapter Forestry Sciences Vol. 52, pages 255–261. KLuwer Academic Publishers,, 1997.

[6] 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.

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

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

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

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