From Discrete Event Simulation to Virtual Reality Environments

Sebastian Nielebock[1], Frank Ortmeier[1], Marco Schumann[2], and André Winge[2]

[1] Computer Systems in Engineering,
Otto-von-Guericke University of Magdeburg, Germany
sebastian.nielebock@st.ovgu.de, frank.ortmeier@ovgu.de
[2] Fraunhofer Institute for Factory Operation and Automation
IFF, Magdeburg, Germany
{marco.schumann,andre.winge}@iff.fraunhofer.de

Abstract. Today's technical systems are often very complex. System dynamics are often hard to predict for humans. However, understanding system behavior is crucial for evaluating design variants and finding errors. One way to cope with this problem is to build logical or virtual simulations. Logical simulations are often very abstract, but can simulate complex behavioral sequences. Virtual reality (VR) simulation is very good for experiencing the system in a view close to reality. However, it is very often static or has only limited dynamics. Until now both approaches exist in relative isolation.

In this paper, we report on our experiences in building a mixed simulation, here a discrete event simulator (DES) is coupled with a virtual reality (VR) environment. We will focus on technical and conceptual challenges, but also present possible use cases for user interaction in this strategy to make more detailed investigations possible. Finally a prototype based on the simulation tool "SLX" and the virtual reality environment "Virtual Development and Training Platform" is used to evaluate the approach.

Keywords: virtual reality, discrete event simulation, synchronization, coupling, model.

1 Introduction

Complex, software-controlled technical systems play a vital role in our modern world. Many processes can't be controlled without them. Nevertheless, this makes it harder for engineers to understand and anticipate the behaviour.

The use of models of the real world system is a well-known strategy to address this problem. There exist a wide variety of models ranging from static, architectural views through formal models including dynamics to precise 3D models. Analysis methods range from best practices through simulation to verification. In this paper, we will focus on 3D models and models expressing abstract behavior. As analysis method, we will only consider (interactive) simulation. Simulation

F. Ortmeier and P. Daniel (Eds.): SAFECOMP 2012 Workshops, LNCS 7613, pp. 508–516, 2012.

has a broad field of applications like in planning of production lines or traffic density[5,9].

Most simulation tools rely on a very abstract representation of physical coordinates and focus on logical places and temporal sequences of events. The common semantic models are discrete event systems (DES). On the other hand, virtual reality simulators typically focus on displaying physical coordinates and orientations very precisely. Temporal evolution is often not possible or implemented in a number of fixed simulation sequences.

As a consequence, each technique itself can give only a limited view on the system: the virtual reality provides understanding of spatial relationships while DES helps understanding temporal understanding. Because of this separation it becomes necessary to model and analyze each of the strategies itself.

The main idea of this paper is to combine the benefits of both in a coupling. So an engineer can use the visualization in parallel to resp. as a front-end of a mathematical DES simulation model. In section 2 it is introduced basic information on "Virtual Reality" and "Discrete Event Simulation" in this paper. The concept of coupling is introduced in section 3 and is evaluated with a prototype presented in section 4. Thereby for illustration, an example system of medium size from logistics scope is considered. Finally the main facts and some future work are summarized in section 5.

2 Related Work

To introduce the topic it is necessary to give an overview about the different techniques and environments and their understanding in this paper. Furthermore some familiar workings are presented.

"Virtual Reality (VR)" is wedded to the definition of a virtual environment. Thereby this environment represents a (mostly) realistic or fictive and visual 3-dimensional presentation generated by a computer. The main element is the user, who controls the actions inside, depending to an underlying control model. The symbioses of visualization and perception of this designed world and its physical properties leads to the term VR [11, p. 3].

In a more technical way the VR-system is a tool to create a set of graphical objects, which are displayed on visualization hardware e.g. a monitor or a CAVE[1]. Using slightly changed images e.g. a different position of an object and play them with a special frame rate[2] leads to the effect of a moving scene, called activities. The state of the objects can be represented as a 6-D-vector: 3 dimensions for the direction and 3 dimensions for the position of the object. There at least 2 special objects: *the path* and *the user*. The path can be used to move an object through the VR, whereby the changing in direction position of the object is automatically done. The user-object is the user interface to interact with the VR like moving through the environment or to grab objects. All graphical information, called *scene* are stored as a *layout*.

[1] Cave Automatic Virtual Environment.
[2] Typically given as displayed frames per second.

The Discrete Event Simulation (DES) is a time discrete simulation whereby in every time step one or more events happen. Concerning to the term of simulation, which is defined as the consideration of a system with its corresponding abstract model over time [7, p. 1], an *event* is timeless appearances like an arrival of ship at a harbor. This means that DES does its calculations on specific points in time and jump to the next event-time immediately. Thereby it is possible that in one calculation multiple events are triggered. In opposite to an event an *activity* is a timed element for instance the discharging of a container from a ship. It is necessary to mention that activities are not part of the DES but of continues consideration of a system.

There have been several approaches concerning to the coupling DES with a VR in the last years. So in [13] there is shown different classifications to describe the type of coupling (table 1).

Table 1. Classification of the coupling[13]

Classification	**Possible implementations**	
Temporal Parallelism	*Concurrent/Online*	*Post-Run/Offline*
Interaction	*Bidirectional*	*Unidirectional*
Hardware Platform	*Monolithic/Homogeneous*	*Distributed*
Visualization Tool Autonomy	*Integrated*	*External*

The categorization "temporal parallelism" distinguishes, whether the DES and the VR are running parallel (concurrent) or sequential (post-run). Another dimension for categorization is how the communication between the both partners is directed. In this case "unidirectional"-case describes that data flows only from the DES to VR. In contrast, the "bidirectional"-case allows both tools to exchange data with each other. Obviously a bidirectional coupling has to be concurrent. Further classifications refer to the used hardware, whether both systems are using a monolithic or distributed platform and the autonomy of the visualization, which characterize whether the DES has an integrated VR or an external one is needed.

Moreover in [13] it is shown that for bidirectional communication messages with timestamps are very useful. Thereby different synchronization methods for those messages are evaluated and a concept of a self-adaptive buffering in the VR system is introduced. Other synchronization methods can be found in distributed simulation, in which conservative and optimistic approaches are used to avoid synchronization faults [4, p. 51ff]. Specific implementations spread from several unidirectional and post-run tools [12,14] to bidirectional solutions[3,10].

3 Concept of the Coupling

The developed coupling may be classified as bidirectional, concurrent, distributed and external. Furthermore the coupling should be created in a way that DES and VR can be run independent (as stand-alone models) from each other. Thereby it consists of 3 parts: The syntactic part (3.1)), which describes time synchronization, the semantic part (3.2)), which describes how semantic information are mapped and the integration of user actions (3.3).

3.1 Temporal Coupling – Synchronization

The main problem to combine the VR and the DES is to synchronize the different views on the time. On the one hand, a DES describes only (discrete) moments in the course of time by events, on the other hand the VR needs a detailed view by representing activities in (fixed) frame rates. In the following, we will assume that time intervals between DES events are much larger - typically several seconds - than those by the VR[3]. Furthermore both, the VR and the DES have a local clock representing their internal simulation time.

During a simulation run the slower clock of the VR runs continuously, while the DES clock has to be delayed between different timestamps. So it becomes possible to visualize activities without interruptions. Thereby all events for a specific point in time have to be calculated within the DES and have to be transmitted to the VR. Concerning to a point in time the events can be communicated either parallel, which means one communication message for all triggered events, or sequential, which means one communication message per event. To reduce the size of a message the sequential strategy will be used whereby the delays caused by communication and calculation effort lead different receiving times. To deal with this problems the messages are constructed in way shown for creating a truck object in figure 1 whereby every event has a specific time stamp and specific instruction names and a list of different arguments. The VR is able to interpret and execute these messages by their instruction names. Hereby the messages will be buffered until the VR clock reaches the point in time expressed by the time stamp[4]. The execution of the messages can be either an event e. g. creating a 3D-object or an activity e. g. moving an object on a path. Because the DES doesn't represent activities, further events have to be delayed until the activity is finished, because they can depend on the execution of this activity. For example a truck has to drive to the loading dock before it can be loaded. Nevertheless parallel executions of activities can be done by using multiple simulation threads, which trigger independent events from each other and are displayed in different VR threads.

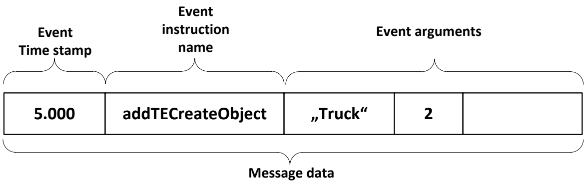

Fig. 1. Structure of a time stamp message

[3] Defined by the frame rate.

[4] It becomes necessary that all events have to be delivered until this point in time is reached.

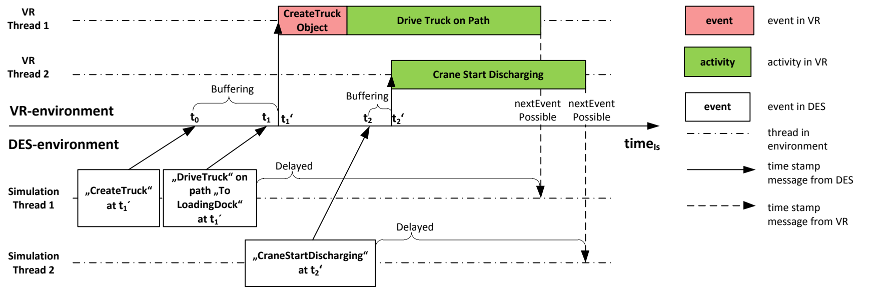

Fig. 2. Communication between DES an VR

A fully communication is shown in figure 2, whereby in this case a truck object is created and drive to the loading dock during a crane is discharging a ship. Hereby the events in the VR executed as events are displayed red and events which are presented as activities are displayed green. At the end of an activity an "a next event possible"-message is triggered with the time stamp to get the next valid time for the simulation thread. Actually these messages can be used to transmit data from the VR to the simulation e.g. measured values by a sensor.

3.2 Spatial Coupling

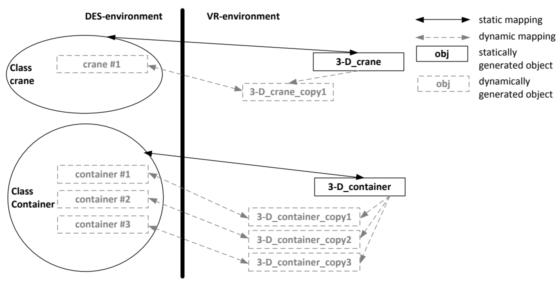

Fig. 3. Class-based semantic coupling

Another challenge is the spatial coupling of the objects in VR and DES. For successful coupling the objects and paths of the virtual reality environment have to be mapped to those in the DES and vice versa. Actually the presentation of the objects is different. A DES sees the objects as logical components, while a VR environment sees them as 3D-polygonal-meshes. To map the objects in a dynamic way a class-mapping is used (see figure 3). Classes of objects from DES (in the example a crane and several containers) are mapped with 3D-objects from the used virtual reality scenario. By doing this it is possible to create or delete meshes in the VR by incarnating or deleting the referred logical objects in the DES. A possible mapping is displayed in 3. However, selecting the mesh is the first step of the mapping. It is also important to set position and orientation. In more general, every state change in the DES must correspond to a (set of) 3D paths in the virtual reality environment. These paths have to be specified manually. However, the current implementation allows easy customization by allocating arbitrary pre-drawn paths to each event. For successful coupling paths and meshes have to couple with each other in the last step.

3.3 User Interaction

The last element in the presented coupling is user interaction. Hereby the knowledge about the system state of the DES resp. the VR has to be considered. The logic controlling the system is within the DES simulation while the VR only provides a visual view on the system. But the user is typically viewing the VR representation. Therefore, it is useful that, users may interact within the virtual reality. For presenting all at a time possible accepted events, the DES must transmit which external events are possible at each time to the VR. User may then select events, which are then passed back to the VR. Of course, in some rare cases update of the list of possible events will conflict with users choices. But this may be solved by informing the user, whenever an invalid events has been passed to the DES.

4 Evaluation

For the test of the approach a real world scenario is used. For evaluating the coupling presented in the previous section a medium sized example of the "Hanse Terminal" in Magdeburg will be used. The "Hanse Terminal" is a harbor for transshipping containers with different loadings from ships to trucks and vice versa. In this logistic process many participants are included e.g. a crane, a reach stacker, trucks and others [8].

The presented example shows a small extract of this logistics process, whereby containers transported by a ship will be stevedored by a crane to be stored on the harbor area. Parallel to this, trucks arrive at the harbor. If containers are stored at the harbor, a reach stacker will load the containers on the trucks. The used DES simulation software tool is SLX [5] by Wolverine Software Corporation [14]. SLX provides an elaborated programming environment to build discrete simulations with parallel threads. As VR-system the VDT-Platform [6] is used. This is a software system that is developed at the Fraunhofer IFF. The VDT-Platform is modular and based on a standardized data model. The architecture of the software system consist three main components [2]

- the Scenario: a certain amount of application specific data that represents the virtual model,
- the Scenario Player: the common set of functionality to handle the Scenario specific data
- the plugin framework: the set of functions for authoring and editing the virtual model and its behavior.

[5] Simulation Language eXtended.
[6] Virtual Development and Training - Platform.

In the past, the platform was primarily developed and used for training of maintenance and service personnel. Meanwhile, the increasing availability of 3D data provides more and more fields of application in which this software frame-work can be used. The main idea is to enable interaction with realistic virtual plants, machinery and products on the basis of immersive 3-D virtual environments. That is the basis of the use of the software framework and the virtual environment in the whole product- and production life cycle [1].

Because of the distributed usability an interface also provided by the Fraunhofer IFF, based on the real-time-interface in [6], is used. This interface uses a shared memory, on which reading and writing is synchronized. is used. This interface uses a shared memory, on which reading and writing is synchronized. The implementation of the VDT-Platform is based on plugins. Thereby a plugin for the offline-simulation, which was still build by the IFF, is extended with an interface to communicate concurrently with the SLX simulation. Furthermore this implement also the concepts of the spatial mapping and the user interface. The SLX model is build with 2 modules, whereby one module displays the simulation model to control the current scene and the other provides general functions to communicate with the VR. In this example the controlled model for the example is created as state model, so that incoming user events can change the internal state and so the behavior of the model.

The implementation was successfully done and showed the desired behavior with the "Hanse Terminal" simulation. So it was possible to visualize the activities triggered by the SLX-simulation in the VR under normal conditions [7] just in time. Multiple user events could be triggered and were correctly processed by the SLX-simulation-model. Nonetheless, the distributed functionality wasn't tested, so that no valid conclusions about this can be made yet.

In opposite to integrated visualization tools like used in Plant Simulation [12] this approach sees the VR and simulation as independent to each other by communicating with a centralized interface. So it was possible to simulate the DES separately to the VR. Another benefit is the logging of events created by the DES. This information can be used to replay special situations in the VR using an offline coupling. Because of the independent design this may be done in a different VR-environment.

5 Conclusion

All in all this paper shows that a coupling of a DES and a virtual environment including time, semantic objects and user interaction is possible and useful. The big benefit however lies in analyzing questions, which may not be answered by the DES alone. For example: How close do trucks and reach stackers pass each other? What quality of service could be expected when monitoring the real world system with a camera system?

The basic concept of the coupling – mapping each event of the DES to an arbitrary set of temporal sequences in the VR environment – is very efficient.

[7] Normal means with a framerate, which leads to a fluid visualization for the user.

Based on the example of the "Hanse Terminal Magdeburg", these techniques are used to build a prototypical implementation of this coupling. This prototype helped us evaluate the usefulness of the coupling as well as performance and feasibility.

In current work further examples are tested to get a better evaluation about the advantages and disadvantages of this coupling. One aspect is setting up a distributed scenario. Much more interesting, however is moving to full co-simulation: for example having the DES simulation react on (continuous time) sensor values modeled in the VR environment only.

References

1. Blümel, E., Fredrich, H., Winge, A.: Applied Knowledge Transfer to European SMEs by Expertise Networks Using Mixed Reality. In: Niedrite, L., Strazdina, R., Wangler, B. (eds.) BIR 2011 Workshops. LNBIP, vol. 106, pp. 90–101. Springer, Heidelberg (2012)
2. Blümel, E., Hintze, A., Schulz, T., Schumann, M., Stüring, S.: Perspectives on simulation in education and training: virtual environments for the training of maintenance and service tasks. In: Chick, S.E., Sanchez, P.J., Ferrin, D.M., Morrice, D.J. (eds.) WSC. ACM (2003)
3. Franke, R.: Kopplung von diskreter Simulation und interaktiver 3D-Visualisierung. Master Thesis, OVGU Magdeburg - Faculty of Computer Science (2004)
4. Fujimoto, R.M.: Parallel and Distributed Simulation Systems. Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, John Wiley & Sons, Inc., Scientific, Technical, and Medical Devision (January 2000)
5. Goettsch, N.: Planung neuer Produktionsaufgaben: Ohne Simulation kein Angebot beim OEM (March 2007), `http://www.it-production.com/index.php?seite=einzel_artikel_ansicht&id=33458` (accessed at March 12, 2012)
6. Kennel, M., Bayrhammer, E.: Eine Schnittstelle zur echtzeitfähigen Kopplung heterogener Simulations-, Steuerungs-, und Visualisierungsapplikationen. In: Forschung vernetzen - Innovationen beschleunigen - 3. und 4. IFF-Forschungskolloquium. Fraunhofer IFF, Magdeburg (April and September 2007)
7. Law, A.M.: Simulation modeling and analysis, 4th edn. McGraw-Hill series in industrial engineering and management science (2007)
8. Magdeburg hafen.de. Hansehafen/GVZ - Hansehafen in Magdeburg-Rothensee (March 2012), `http://www.magdeburg-hafen.de/magdeburg-hafen/mdhafen.html` (accessed at March 10, 2012)
9. Planung Transport Verkehr AG (2012), `http://www.ptv.de/software/verkehrsplanung-verkehrstechnik/software-und-system-solutions/vissim/` (accessed at March 12, 2012)
10. Raab, M.: Mechanismen zur Interaktion zwischen virtuell-interaktiver 3D-Umgebung und echtzeitfähiger Ablaufsimulation. Master Thesis, OVGU Magdeburg - Faculty of Computer Science (March 2007)
11. Schumann, M.: Architektur und Applikation verteilter, VR-basierter Trainingssysteme. PhD thesis, Faculty of Computer Science - OVGU Magdeburg, Fraunhofer IFF, Magdeburg (February 2010)

12. Siemens PLM Software Inc.: Plant Simulation Simulation, Visualisierung, Analyse und Optimierung von Produktions- und Logistikprozessen, http://www.plm.automation.siemens.com/de_de/Images/plant_simulation_tcm73-62431.pdf (accessed at February 18, 2012)
13. Straßburger, S., Schulze, T., Lemessi, M., Rehn, G.D.: Temporally parallel coupling of discrete simulation systems with virtual reality systems. In: WSC. ACM (2005)
14. Wolverine Software Corporation. Wolverine Web (February 2012), http://www.wolverinesoftware.com/ (accessed at February 8, 2012)