Distributed Traffic Simulation based on the High Level Architecture

Ulrich Klein
Thomas Schulze
Steffen Strassburger
Department of Simulation & Graphics (ISG)
Department for Computer Science
Otto-von-Guericke-University Magdeburg
Universitaetsplatz 2
D-39106 Magdeburg, Germany
{uklein; tom; strassbu}@isg.cs.uni-magdeburg.de

Hans-Peter Menzler
Competence Center Informatik (CCI) GmbH
Lohberg 10
D-49716 Meppen, Germany
menzler@cci.de

Keywords:
Traffic Simulation, High Level Architecture, Distributed Simulation,
Interoperability, Reusability, Legacy Applications, RTI Performance Issues, VRML

ABSTRACT: *Although originating from the military simulation domain, the High Level Architecture also addresses the needs of the civil simulation community asking for software interoperability and reusability. As a joint industry/academic effort, a traffic simulation prototype has been developed by CCI and ISG which is composed of heterogeneous software components. It serves as a testbed for interoperability and performance issues and will provide a framework for future enhancements. The prototype, its federates and the development process are described as well as the special viewpoint of the civil simulation community. A summary of lessons learned and ongoing work concludes the paper.*

1 INTRODUCTION

Interoperability and reusability are two key topics for the military and civil modeling and simulation community. In this regard, the High Level Architecture as the newest technology for distributed simulation-based systems is a big step forward.

In order to evaluate the potential of the High Level Architecture in the civil application domain, the Institute for Simulation and Graphics (ISG) tested several simulation and animation tools for HLA compliance. Requirements have been compiled and experiences have been made from this first step.

The Competence Center Informatik GmbH (CCI) in its role as an advisory board for the german federal office for defense technology and procurement (BWB) evaluates approaches like HLA and is experienced in real-time simulation.

As a joint effort by the ISG and CCI, we developed a prototype federation covering main aspects of interoperability and the federate development process. A driving simulation scenario was chosen to be our civil prototype application area.

2 RATIONALE OF WORK

2.1 The High Level Architecture

The HLA addresses interoperability and reusability of simulation and other components which are well-known needs of the military and the civil simulation domain as well [1].

In order to act as an HLA federate, software has to follow the ambassador paradigm which results in the need for

- linking a software library, and
- implementing a federate ambassador.

In addition, depending on the structure and type of simulation to be enhanced for HLA, alterations of the time management and / or event handling algorithms have to be made.

Therefore the integration effort varies heavily for legacy applications, commercial tools, etc.; an open software / tool architecture and the availability of source code are two of the factors which influence the ease of integration to a large extent.

2.2 HLA in the civil application domain

In the civil simulation and animation (S&A) community software tools are widely used. In order to use these tools under HLA, they have to be extended to become HLA compliant which results in a different starting point compared to the extension of source code of a general programming language like C++, Ada95 or Java.

Like in the military domain, numerous problems and models exist which would benefit enormously from being modularized, distributed or networked via their underlying simulation tools (now being interoperable).

In a first step, the requirements to which tools have to comply with have been derived [4]. Because of the stand-alone, autonomous nature of today's S&A tools these requirements proof to be a challenge for some tools as our tests have shown.

We have tested simulation tools like GPSS/H, JavaGPSS and SLX, the animation tools Proof Animation and Skopeo, as well as other software systems.

2.2.1 Simulation Tools

The main requirements for the integration of simulation tools into the HLA (here formulated for event driven simulation tools) are the accessibility of the event list management (in order to interleave internal and external events) and the transparent access to simulated objects modeled within the tool.

The degree to which syntax extensions are necessary to cover HLA functionality is another crucial factor; the more the operation within a distributed environment under HLA is hidden from the user and, related to that, the easier the migration of existing models is, the higher the acceptance of the tools will be.

The new simulation tool SLX (Simulation Language with eXtensibility, the successor of GPSS/H) was successfully extended to support HLA-aware simulation models [2]. SLX runs on Windows 95 / NT machines and uses an external model-independent Wrapper-Library written in C / C++, which in turn links the RTI library provided by the HLA software package [9].

2.2.2 Animation Tools

As a natural complement to pure simulation tools, animation and visualization tools are needed. While animation systems are partly designed for trace-file based post-run animation and some for online-animation, the (main) source of animation information under HLA is the RTI.

Since the characteristics of the federation to be animated and the user needs may vary significantly, a concept for HLA-based animation has been derived and partly been implemented using the animation system Skopeo [6, 7].

2.2.3 Additional Building Blocks

In addition to the simulation and animation tools which are enhanced to operate under HLA, additional building blocks for civil and emergency management applications have been identified and integration concepts formulated, like online- / real-time information (e.g. environmental information systems), geographic information systems (GIS) and other "timeless" systems [3].

2.3 Distributed Traffic Simulation

The ISG has a long experience in traffic simulation, resulting in a wide variety of simulation models suited for different traffic management tasks.. CCI has extensive experience within the real-time simulation domain, e.g. vehicle training simulators. Therefore we have chosen the traffic simulation domain to be the application area for our first prototype.

3 THE DISTRIBUTED DRIVING FEDERATION

3.1 General Design Aspects

The overall intention was to bring heterogeneous federates from different simulation domains together to form a distributed driving simulation within a virtual environment (fig. 1).

The federation consists of a *Traffic Simulation* federate designed to efficiently model urban street traffic flow on a detailed microscopic level. This model operates on a virtual street network and simulates cars resp. their drivers.

Cars are also brought into the federation by the second *Driving Simulation* federate, which is a man-in-the-loop simulator allowing a person in front of the simulator to steer the car in real-time. The interactions between the cars are what the federation focusses on.

A third *Visualization* federate is used as an observer especially for the site that runs the traffic simulation which has no visualization component.

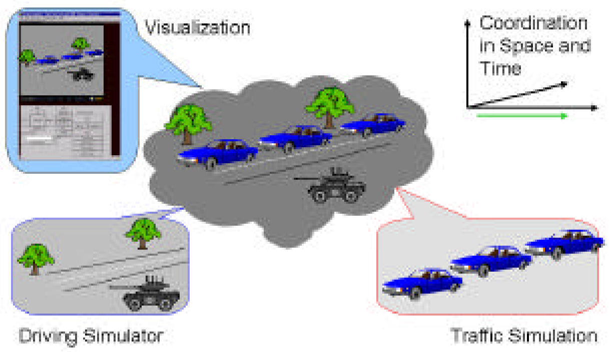

Fig. 1: The objects each of the federates is modeling and how they are combined in the common event space of the federation execution

A virtual environment of approximately 4 km by 4 km was chosen to include a street running straight in east-west direction. Although the federates are capable of modeling crossings, traffic lights, more complex traffic patterns etc., there was no need for added complexity because of the main focus on basic interoperability issues.

Main focus of the prototype federation is on time management interoperability (real time vs. discrete event oriented), platform interoperability (SGI vs. Windows PCs vs. platform independent Java virtual machine), language interoperability (C++ vs. SLX vs. Java), independent federate development based on a common FOM definition and (at a later stage) network interoperability (LAN vs. ISDN-Link vs. Internet).

3.2 Development Process

After initial definitions of the FOM and the SOMs, the federates were developed independently at the CCI and ISG sites. Thereby we were able to evaluate the distributed federate development style supported by the HLA philosophy.

Whereas the CCI federate was almost built from scratch, the ISG traffic simulation used a legacy SLX simulation model intended to exactly mimic the behavior of car drivers following other cars within an HLA environment.

3.3 Object Model

As basis for the Federation Object Model the Real-time Platform Reference FOM (RPR FOM) [12] was chosen. Because of the restricted scenario modeled within the prototype federation, it was sufficient to use only a small generic part of the RPR FOM (the *BaseEntity* and the *Collision* classes as shown in Table 1 and 2). Only the enumerated datatype "*EntityKind*" was adopted to the enumeration {*Tank*, *Car*, *Truck*}.

Table 1: Objects and their attributes, publishers (owners) and subscribers

Object Class	Attribute	Published by	Subscribed by
BaseEntity	as specified in RPR FOM 0.1.7	Driving Simulation, Traffic Simulation	Driving Simulation, Traffic Simulation, Visualization

Table 2: Interactions sent between federates

Interaction Class	Parameter	Sent by	Received by
Collision	as specified in RPR FOM 0.1.7	Driving Simulation	Traffic Simulation

The collision detection is performed by the Driving Simulation.

3.4 Hardware Configuration

The driving simulation is a C++ application using the SGI Performer software on a SGI Onyx machine. The traffic simulation is a SLX simulation model running on Wintel platforms.

The visualization federate Skopeo needs only a Java virtual machine and, optionally, a VRML2.0 plugin; currently, the SGI CosmoPlayer for Windows NT is used.

The driving simulation at the CCI site was connected to the other machines within the University of Magdeburg LAN with an ISDN 64 kbps dial-up connection.

Table 3 summarizes the technical environment for the federation; we used the DMSO RTI software version 1.0 release 3 for the IRIX, Win32 and Java platforms. In most cases, the ISG site also hosted the RTIExec and Fedex processes.

Table 3: Overview of Federates and their time management characteristics

Federate	Site	Platform/ Operation System	Language/ Software
Driving Simulation	CCI	SGI Onyx Risk 4400 200 MHz, 128 MB RAM / IRIX 6.2	C++ Performer 2.2
Traffic Simulation	ISG	double Pentium II 266 MHz, 128 MB RAM / NT4.0	SLX HLA-SLX-Library
Visualization Federate	ISG; location indep.	Java Virtual Machine	Java VRML2.0 plugin
RTIExec	ISG	NT4.0	C++
FedEx	ISG	NT4.0	C++

3.5 Time Management Interoperability

The interoperability of different time management schemes was one of the main aspects of the prototype federation.

The driving simulation is a real-time DIS-like simulation, while the traffic simulation uses a classic discrete event-oriented time advancement approach. While the driving simulation doesn't require causality for proper operation, the traffic simulation relies on strict causality. The visualization federate also requires causal event ordering.

In the HLA terminology, the driving simulation was not time constrained, but time regulating; the time stamp ordered events generated by the driving simulation were then used to pace the traffic simulation which is time constrained and time regulating. The visualization federate is time constrained, but (because of its passive role) not time regulating.

The visualization federate can also be set to be time-regulating in order to force other (time-constrained) federates to wait for it. Use cases include

- if the visualization federate is used as an user interface when the current federation status is to be displayed;
- if future interactive capabilities via the VRML2.0 interface are used.

Table 4 summarizes the time management characteristics of the federates.

Table 4: Overview of Federates and their time management characteristics

Federate	Time constrained	Time regulating	Time advance
Driving Simulation	No	Yes	Real-time
Traffic Simulation	Yes	Yes	Event oriented
Visualization Federate	Yes	Yes / No	scaled real time

4 THE DISTRIBUTED DRIVING FEDERATES

4.1 The Driving Simulation

The driving simulation has been built nearly from scratch reusing only a 3D database which was originally developed as a virtual environment for a tank commander training simulation.

A human driver should be able to steer the tank / truck within the terrain as well as on the road, interacting with other cars on the street; the mouse is used for control.

The FOM as a first stage in the federation development process (FEDEP) equals the SOM for both, the traffic and the driving simulation. Only a few number of functions from the RTI 1.0-3 have been used so far to support the SOM, which itself is based on the well known Real-time Platform Reference-FOM [12]. Those functions stem from the essential object-declaration, object-management and time-management services of the RTI. Additionally, an interaction class Collision has been introduced.

It is worth noting that the driver software-architecture divides the application into two almost independent domains, i. e. the virtual simulator (driver) and its HLA interface. An object-oriented design promotes highly flexible structures to keep even the first prototypes as a base for future extensions and/or improvements. The method of choice is to introduce an Intra-SOM-layer (ISOM) separating the interface from the simulator and at the same time being the internal representation of the SOM (without referring to any HLA interface functions of the RTI). This allows both the interface and the simulator to access a shared memory portion of the ISOM-layer based on commonly agreed ISOM classes [5]:

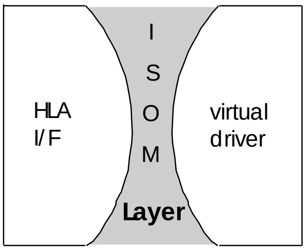

Fig. 2: intra-SOM-layer approach of the driving simulator

The virtual driver was developed in C++, based on SGI-Performer 2.2. The ISOM-layer mainly consists of object-references which mimic the above mentioned read and write memory shared by the HLA-interface and the driver. Several timeout events used to tick the RTI and to dead reckon entities use the X-Window's event mechanisms.

4.2 The Traffic Simulation

The traffic simulation federate is based on a traffic simulation model which was developed to model the complex psycho-physical behavior of car drivers following other cars [8]. It was written in an event-oriented manner using an early version of the SLX simulation language and can be adopted to complex traffic infrastructures including crossings, traffic lights, pedestrians etc. Originally it was used to disprove the belief that this problem can only be solved efficiently by time-stepped algorithms.

This legacy simulation was ported to the current version of SLX which we had already enhanced for HLA. In order to reflect the new role as cooperating federate in a federation, only minor changes of the model were necessary.

A more detailed description of SLX and the HLA-extension of SLX can be found in [9, 10].

4.3 The Animation System

The animation tool Skopeo started as a java-based Proof Animation compatible 2D post-run animation tool and was recently enhanced for 3D visualization using VRML 2.0 [6, 7]. After the simulation has generated a trace-file, Skopeo used it for animation mainly in scaled real-time depending on the user's preferences.

It was successfully extended as an HLA federate and is now able to visualize ongoing federation activity online (fig. 4 and 5) [7].

The beta-version of the Java RTI software from DMSO was used to connect to the runtime infrastructure, making use of the wire compatibility of the C++ and Java software.

In order to be a flexible visualization tool for different kinds of federations ranging from real-time to as-fast-as-possible and time-stepped, a flexible time management concept for Skopeo was developed which decouples federation time and visualization time. Therefore a client/server-architecture was introduced.

Incoming events are stored in a trace cache and, regarding the role of Skopeo within the federation, time advances as fast as the RTI allows the time-constrained Skopeo to. This is achieved by the Skopeo Workgroup Server which also has the role of the federate in the HLA sense.

On the visualization side, information is retrieved from the trace cache in a manner appropriate to the user's need for smooth animation, VCR-like playback functionality up to always-up-to-date online visualization.

This is achieved by gradually coupling federation and visualization time advances. In addition, the concept also allows the management of optimistic events which may be implemented in a future version of Skopeo.

The visualization is done by Skopeo clients which reside as applets in web browsers. In order to allow multiple users to visualize independently and to keep the network and RTI load at a minimum, only the Skopeo workgroup server is a federate feeding the trace cache. The client/server- and inter-client communication is performed using CORBA. The concept is shown in figure 3.

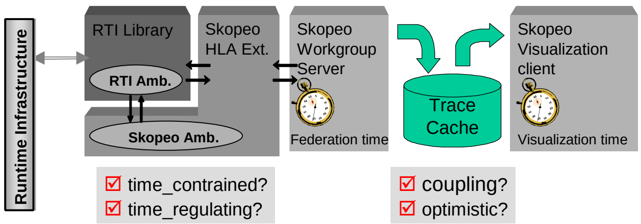

Fig. 3: Skopeo time management concept

Work is underway to use the VRML2.0 based 3D visualization for interaction with HLA objects, resulting in an active role of Skopeo; users will be able to touch, query, modify and move objects modeled within the federation.

Currently, the HLA-version of Skopeo has been successfully tested in several small prototype federations at ISG; it is planned to become a federate of the distributed driving simulation in the near future as part of a new federation execution cycle.

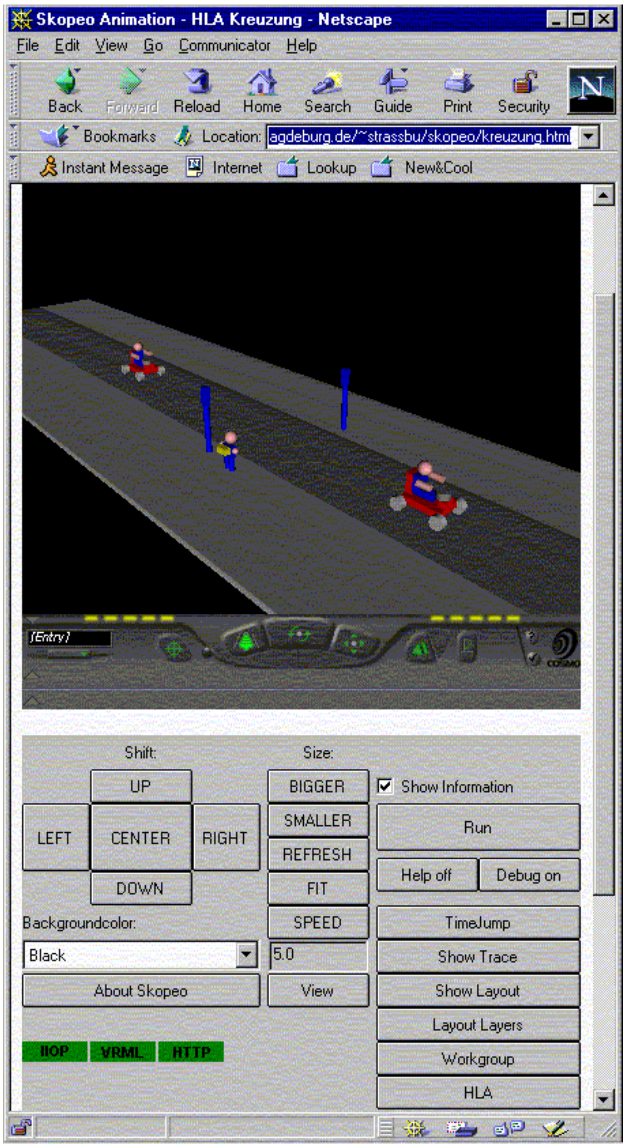

Fig. 4: Skopeo 3D online visualization of the current federation status via VRML 2.0

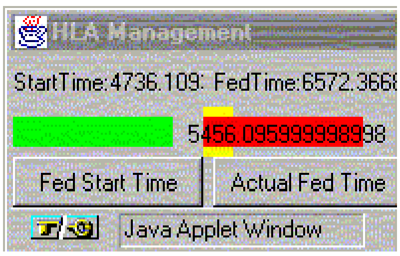

Fig. 5: Skopeo HLA time management control for federation time / visualization time coordination

Figure 4 shows a screenshot of the Skopeo visualization system while it is monitoring objects of a small traffic simulation prototype used at ISG. Figure 5 depicts the Skopeo time management window which is used to control federation / visualization time advance.

5 FEDERATION EXECUTION

Several online sessions have been conducted to run the federation with different parameter sets and camera viewpoints.

The standard procedure for the truck driver was to start somewhere off-road, move to the first parking space, and wait for a gap to enter the street (see fig. 6). Then, the truck moved along the street until it reached the second parking lot in order to leave the street.

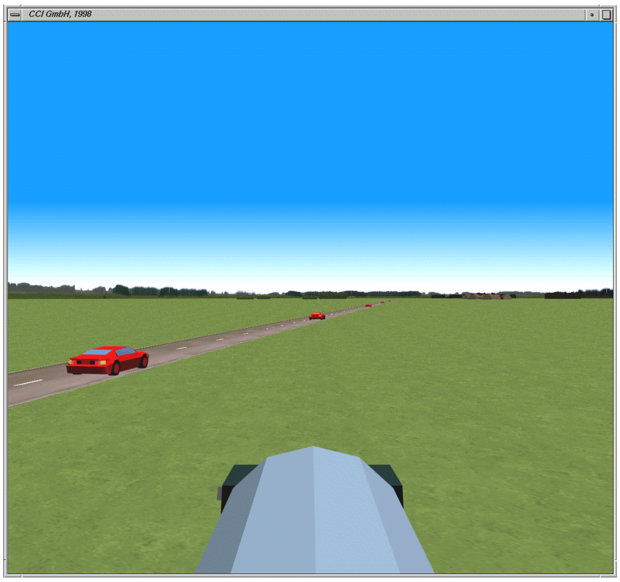
Fig. 6: Front view by the truck driving off-road; the red Ferraris are modeled by the traffic simulation. Visualization by the driving simulator

Due to their higher speed, a growing number of cars compiled behind the truck in a realistic manner. The car drivers were told not to overtake the truck. As soon as the truck has left the street, the cars begin to accelerate to their initially assigned, randomly generated individually desired speeds.

This kind of object interaction can be seen in the figure sequence 7, 8 and 9 showing the truck entering the street (fig. 7), driving down the street (fig. 8) and leaving the street (fig. 9).

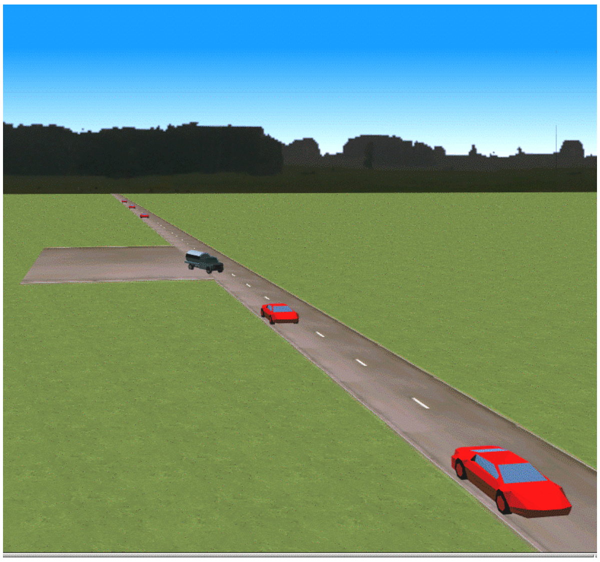

Fig. 7: The truck is entering the traffic flow

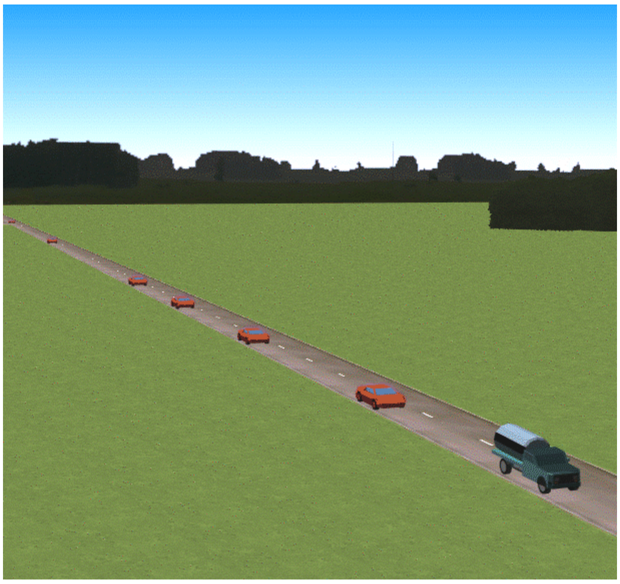

Fig. 8: The truck driving down the street

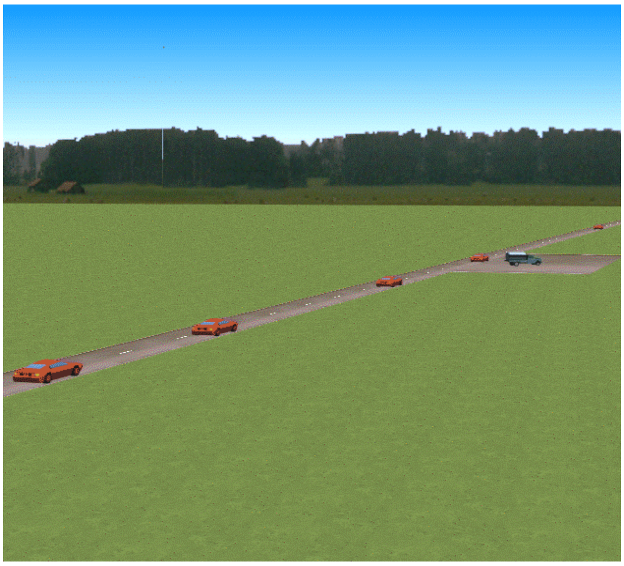

Fig. 9: Bird‘s eye view of the truck leaving the street

6 PERFORMANCE ISSUES

With minor modifications of the parameters of the traffic simulation (intensity and variability of car traffic), a wide range of RTI load could be generated. Given our federate time advance characteristics, discontinuities would show up as delayed messages from the Traffic Simulation, resulting in delayed or false information seen by the "truck driver".

Because of the moderate car traffic density and the simple environment we have used in our first test series, we have noticed no causality glitches as the time constrained discrete event traffic simulation was able to keep up while being paced by the driving simulation.

The performance of the traffic simulation without the driving simulation being present in the federation (therefore running as-fast-as-possible) suggests that either much more complex simulations could be run on the platform used or hardware with minor performance could be used.

Vice versa, our first experiences with the HLA-enabled web-based VRML2.0 3D visualization tool Skopeo show that sophisticated graphic hardware is needed in order to have acceptable frame rates for animated VRML2.0 worlds; it is worth noting that the overhead of

1) being an HLA federate and receiving the animation data from the RTI and
2) using CORBA as communication platform between the Skopeo Server (which is also the federate) and the visualization clients (the applets)

is minimal compared to the performance needed for the graphics. Nevertheless, this is work-in-progress and more information will be published in the future.

Performance experiments has been conducted by CCI based on a federation consisting of multiple driving simulations [5]. Latency and bandwidth experiments with the full federation will be conducted in the near future.

7 SUMMARY AND OUTLOOK

The experiences with the prototype showed that HLA is well suited for the military and civil domain. Especially the interoperability of different time management schemes commonly used in the civil simulation community offer a new flexibility which was previously not available.

Other prototypes focussing on HLA-based manufacturing systems [10] underpin the importance of the HLA in civil application areas.

REFERENCES

[1] Defense Modeling and Simulation Office (DMSO). "The High Level Architecture Homepage." http://hla.dmso.mil 1998.

[2] Henriksen, J. O. "An Introduction to SLX." In Proceedings of the 1997 Winter Simulation Conference, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, pp. 559-566, 1997.

[3] Klein, U. "Distributed Simulation for Emergency Management based on the High Level Architecture." Proceedings of the 5th conference of the International Emergency Management Society (TIEMS'98), May 19-22, Washington, D.C. 1998.

[4] Klein, U. and S. Strassburger. "The High Level Architecture: Requirements of interoperable and reusable simulations by example of traffic and infrastructure simulations" (german). Proceedings of the 11th Simulation Symposium ASIM 97, Nov. 11-14, Dortmund, Germany 1997.

[5] Menzler, P., and G. Hopmann. "Merging HLA with a Virtual Simulator." Paper to be presented at the I/ITSEC 1998.

[6] Ritter, K. C. "The Skopeo Animation System." Available online at http://simos2.cs.uni-magdeburg.de/Skopeo/Ani.html 1998.

[7] Ritter, K. C., U. Klein, S. Strassburger and M. Diessner. "Web-based animation of distributed simulations using the High Level Architecture" (german). Proceedings of the Simulation & Visualization Conference Magdeburg, March 5-6, SCS Europe, pp. 41-52, 1998.

[8] Schulze, T. and T. Fliess. "Urban Traffic Simulation with psycho-physical vehicle-following models." In Proceedings of the 1997 Winter Simulation Conference, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, pp. 1222-1229, 1997.

[9] Strassburger, S. and U. Klein. "Integration of the simulation tool SLX into the High Level Architecture." (german). Proceedings of the Simulation & Visualization Conference Magdeburg, March 5-6, SCS Europe, pp. 32-40, 1998.

[10] Schumann, M., E. Bluemel, T. Schulze, and S. Strassburger. "Using HLA for factory simulation." Paper 98F-SIW-060 presented at the Simulation Interoperability Workshop Fall 1998, Orlando, Florida 1998.

[11] Strassburger, S., T. Schulze, U. Klein, and J. O. Henriksen. "Internet-based Simulation using Off-the-Shelf Simulation Tools and HLA" to be presented on the Winter Simulation Conference 1998, Washington, D.C. 1998.

[12] Simulation Interoperability Standards Organization (SISO), http://www.sisostds.org/stdsdev/rfom, 1998.

AUTHOR BIOGRAPHIES

ULRICH KLEIN is a Ph.D. candidate at the University of Magdeburg, Germany. He holds a master's degree in Industrial Engineering from the University of Karlsruhe and has been involved in Emergency Management since 1992. He has two years of experience as Project Manager for Command, Control, and Communication Systems for Public Safety and Security in Europe. His research topics include Emergency Management, Urban Infrastructure Management and Logistics, Geographic Information Systems, and the High Level Architecture.

HANS-PETER MENZLER studied theoretical physics at the University of Osnabrueck, Germany, and received his Ph.D. in mathematical and numerical studies on optical waveguides. He worked as a scientist at the Max-Planck-Institute for Plasmaphysics, Garching and became a system engineer at Competence Center Informatik GmbH (CCI), Meppen, Germany, in 1992. Since 1996 he has been working in the training and simulation department of CCI. His work focuses on the object-oriented development of interactive graphical user interfaces and planning algorithms in manufacturing and emergency management areas.

THOMAS SCHULZE is an Associate Professor at the Otto-von-Guericke-University Magdeburg in the Department of Computer Science. His research interests include modeling methodology, public systems modeling, traffic simulation, and distributed simulation with HLA.

He is an active member in the ASIM, an organization for simulation in Germany.

STEFFEN STRASSBURGER is a Ph.D. student at the Simulation and Graphics Department in the Department of Computer Science, Otto-von-Guericke-University, Magdeburg. He holds a master's degree in computer science from the same university. His experience with inter-networking and simulation includes a one-year-stay at the University of Wisconsin, Stevens Point. His main research interests lie in distributed simulation and the High Level Architecture.