

# **Demonstrationen von HLA-basierten verteilten Simulationsmodellen**

## **Demonstration of Simulation Models based on HLA**

H. Tietje, CCI Meppen

S. Straßburger, Universität Magdeburg

U. Klein, FhG-IFF Magdeburg

**Zusammenfassung:** *Mit der Verfügbarkeit der entsprechenden Basissoftware (RTI) begann an der Universität Magdeburg 1996 die Realisierung verteilter Simulationsmodelle auf der Basis der High Level Architecture (HLA). Der Beitrag beschreibt drei unterschiedliche Projekte, die gemeinsam von verschiedenen Partnern erarbeitet wurden. Mit diesen Projekten wurden Möglichkeiten der Wiederverwendbarkeit von existierenden Simulationsmodellen evaluiert und neue Strukturen hinsichtlich der Interoperabilität zwischen heterogenen Simulationsmodellen entwickelt.*

**Summary:** *With the availability of the HLA Runtime Infrastructure Software in 1996 the University of Magdeburg started the implementation of distributed simulation models based on the High Level Architecture (HLA). This paper describes three different projects which were realized with different partners. These projects were used to test the possibilities of re-using existing simulation models and to develop new approaches for interoperability between heterogeneous simulation models.*

### **1. Motivation**

In dem Vortrag werden drei auf der High Level Architecture (HLA) basierende verteilte Simulationsprojekte vorgestellt, die gemeinsam zwischen den beteiligten Partnern entwickelt wurden. Ziel des Vortrages ist die Demonstration, daß

- die HLA-basierte verteilte Simulation die Grenzen von Universitätslaboratorien überwunden hat,
- die Interoperabilität zwischen und die Wiederverwendbarkeit von Simulationsmodellen eine neue Qualitätsstufe erreicht haben und
- neue Applikationsgebiete für die Simulation erschlossen wurden.

### **2. Projekt Interaktiver Fahrsimulator**

Diese Federation wurde von der Otto-von-Guericke-Universität Magdeburg (UniMD) gemeinsam mit dem Competence Center Informatik GmbH Meppen (CCI) entwickelt [6,7]. Die beteiligten Partner fügten unabhängig voneinander entwickelte

Federates in dieser Federation zusammen. Die Simulationsfederates basieren auf existierenden Simulationsmodellen, die hinsichtlich ihrer HLA-Einbindung erweitert werden mußten. Diese Erweiterungen bezogen sich auf folgende Aspekte:

- Kommunikation mit der RTI,
- Interoperabilität zwischen den Modellen,
- Synchronisation des lokalen Zeitfortschritts.

Die beteiligten Federates werden im folgenden kurz charakterisiert:

- Federate Fahrsimulator: Dieses Federate wurde auf der Basis eines existierenden Echtzeit-Fahrsimulators für Geländefahrzeuge entwickelt. Ein menschlicher Fahrer fährt in diesem Simulator ein Geländefahrzeug in einer synthetisch erzeugten Umgebung, die keinen Straßenverkehr modellieren kann. Der Simulator ist in C++ für Unix-Maschinen implementiert und offeriert eine 3D-Visualisierung auf einer Silicon Graphics Maschine. Dieses Federate wurde von CCI entwickelt.
- Federate Verkehrssimulator: Dieses Federate wurde an der UniMD entwickelt. Es basiert auf einem existierenden ereignisorientierten Straßenverkehrssimulator, dessen Schwerpunkt auf der Abbildung des psycho-physischen Fahrzeugfolgeverhaltens liegt. Dieser Simulator wurde in der Sprache SLX implementiert und läuft auf Standard Intel PCs unter Windows NT [1]. Mit diesem Federate wird Straßenverkehr nachgebildet. Die Fahrzeuge haben im Zusammenspiel mit dem Federate Fahrsimulator auf Aktionen des Fahrers zu reagieren. Im Gegenzug muß der Fahrer im Federate Fahrsimulator auf die jeweiligen Fahrzeuge aus dem Straßenverkehr reagieren.

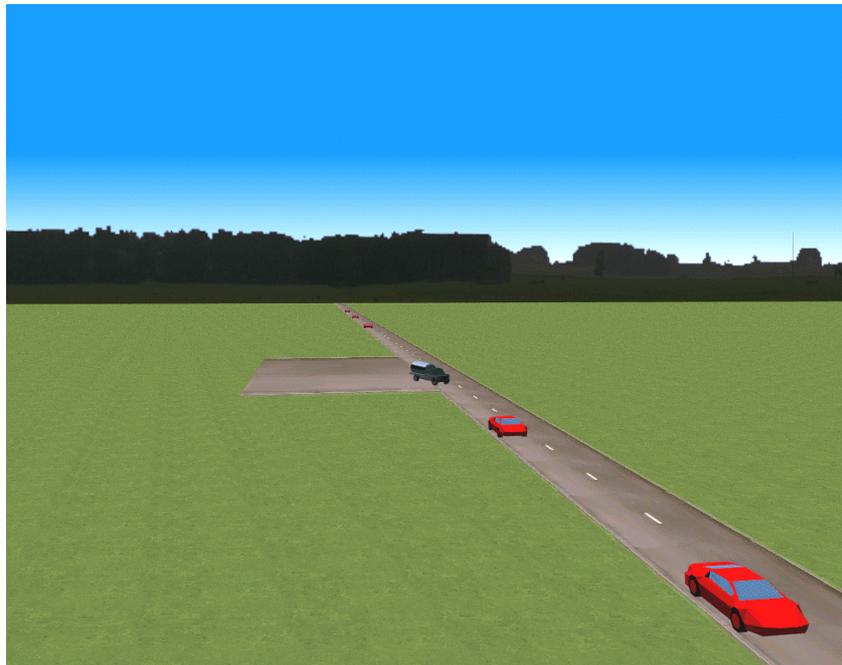


Abbildung 1: Blick auf die virtuelle 3D-Umgebung

Die beiden Simulationsfederates mußten ihre alte Funktionalität erweitern. Der Fahrsimulator hat zusätzlich die Fahrzeuge auf der Straße zu visualisieren. Abbildung 1 zeigt einen Blick auf die virtuelle 3D-Umgebung.

Der menschliche Fahrer muß nur auf die Fahrzeuge auf der Straße reagieren, wenn er mit seinem Fahrzeug den Straßenverkehr beeinflusst. Zur selben Zeit hat auch das Straßenverkehrsmodell auf den externen menschlichen Fahrer zu reagieren. Beide Federates operieren nun in einem neuen gemeinsamen Ereignisraum. Sie müssen ihre Zustände in Raum und Zeit koordinieren. Zur Umsetzung dieser Interoperabilität ist HLA ein geeigneter Rahmen. Die Federation Verteilte Fahrsimulation zeigte, daß eine unabhängige Entwicklung von Federates und eine Wiederverwendung von existierenden Simulationsmodellen als Federates möglich ist.

### **3. Kopplung von SLX und Simplex3**

Es wird gezeigt, wie SIMPLEX3 und SLX als zwei grundsätzlich verschiedene Simulationssysteme mittels des HLA-Standards gekoppelt werden können. Als Anwendungsbeispiel wird das Modell Abfüllanlage beschrieben, das aus einem zeitkontinuierlichen und einem zeitdiskreten Modell besteht. Beide Modelle arbeiten mit Hilfe des HLA-Standards zusammen und simulieren ein Modell einer verfahrenstechnischen Abfüllanlage für Fässer. Dieses Modell enthält neben der Logistik auch einen kontinuierlichen Prozeß und zeigt damit, daß unter HLA selbst unterschiedlichste Simulationssysteme miteinander gekoppelt werden können [5].

SLX ist ein sehr schnelles Simulationssystem für diskrete Simulationsmodelle, während unter SIMPLEX neben diskreten auch kontinuierliche Modelle ausführbar sind. Ebenfalls erfolgt die Modellspezifikation und die HLA-Anbindung in beiden Systemen aus Nutzersicht auf völlig verschiedene Arten. Hierdurch wird die Möglichkeit eröffnet, Teilmodelle von denjenigen Simulationssystemen bearbeiten zu lassen, welche hierfür am geeignetsten sind.

Das SLX-Modell simuliert die logistischen Prozesse in einem Transportunternehmen, eine Aufgabe, für die SLX durch seine prozessorientierte Weltsicht bestens geeignet ist. Das Simplex-Modell stellt die Vorgänge in einer chemischen Abfüllanlage dar, wobei der Füllprozeß über Differentialgleichungssysteme abgebildet wird. Da Simplex 3 eine Vielzahl von numerischen Integrationsverfahren anbietet, ist Simplex für diese Modellierungsaufgabe prädestiniert. Mittels HLA ist es nun möglich, Teile des Gesamtmodells mit dem dafür am besten geeigneten Simulator zu entwickeln und als ein Gesamtmodell agieren zu lassen. Im Gesamtmodell erfolgt eine Generierung von Aufträgen für die Abfüllanlage und der anschließende Transport im Logistikteil (SLX), während das Simplex-Modell die Auftragsbearbeitung durchführt.

Zur Online-Animation des Gesamtsystems wurde Proof Animation verwendet. Proof Animation ist ein allgemeines Animationssystem für die Windows-Welt, welches von beliebigen Programmen zur Online- und Postrun-Animation verwendet werden kann. In der entwickelten Referenzanwendung wurde SLX mit dem Logistikmodell gleichzeitig zur Erzeugung der Animation mittels Proof Animation verwendet. Abbildung 2 zeigt das daraus resultierende Setup der Federation.

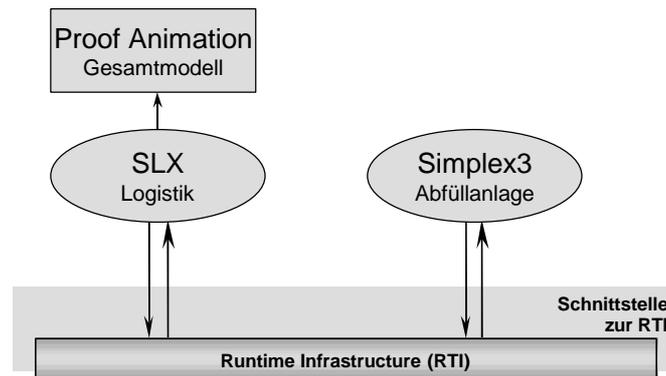


Abbildung 2: Setup der Simplex-SLX-Federation

Die Durchführung der Animation durch SLX ist jedoch keine grundsätzlich notwendige Herangehensweise. So wäre es z.B. auch möglich, Simplex 3 die Animation durchführen zu lassen oder ein separates Federate zur Ansteuerung von Proof Animation zu verwenden. Es ist lediglich zu beachten, daß das Federate, welches die Animation „produziert“, sämtliche für die Federation relevanten Visualisierungsinformationen empfängt. Einen Screenshot der für das Referenzbeispiel durchgeführten Visualisierung zeigt Abbildung 3.

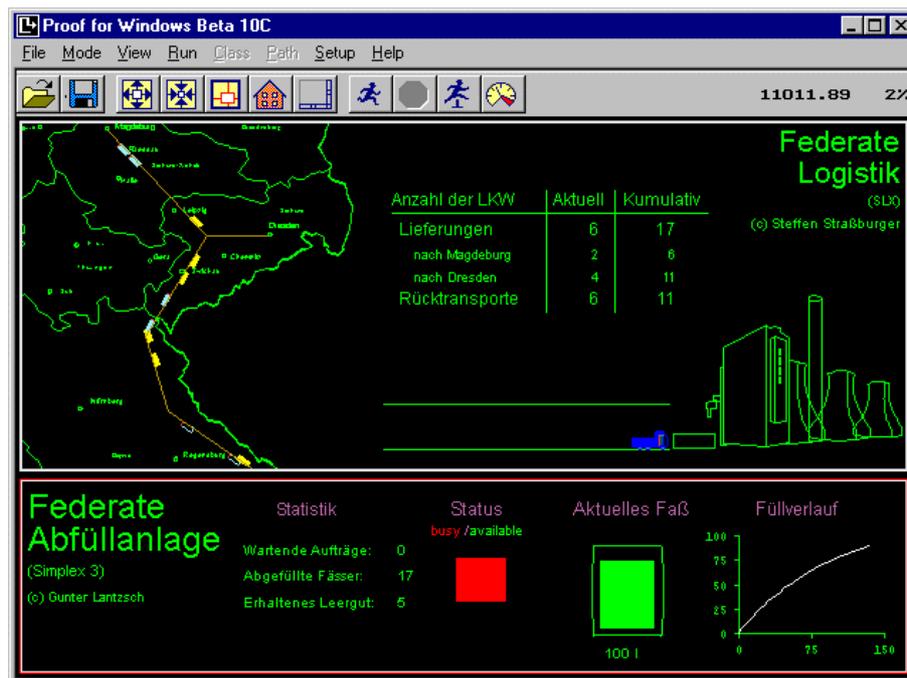


Abbildung 3: Screenshot der Simplex-SLX-Federation

Die Entwicklung dieses Prototyps hat gezeigt, daß HLA Möglichkeiten bietet, unterschiedliche heterogene Simulationsmodelle zu koppeln. Die Heterogenität bezieht sich sowohl auf das Simulationssystem zur Modellerstellung, als auch auf

den Modellierungsansatz. Bisherige Ansätze zur Kombination von diskreten und kontinuierlichen Modellteilen sind charakterisiert durch die Verwendung von monolithischen kombinierten Simulationssystemen oder von auf die erstellten Simulationsmodelle angepassten Kommunikationsprotokollen. Mit der Verwendung von HLA wird eine neue Stufe in der Kopplung zwischen diskreten und kontinuierlichen Modellen erreicht.

#### 4. Integration von Online-Daten in Simulationsmodelle

Die Verarbeitung aktueller Meßwerte ist Grundlage für eine Vielzahl von Anwendungen von Leitständen, Umweltinformationssystemen, Flottenmanagement bis hin zu human-in-the-loop-Trainingssimulatoren. Dabei werden sehr unterschiedliche Anforderungen an die Aktualität der Onlinedaten gelegt, die von harten Echtzeitbedingungen im Millisekundenbereich bis hin zum Minutenbereich reichen können. Verteilte heterogene Systeme mit Simulationsfähigkeiten können ebenfalls von der Einbeziehung von Echtzeit-/Online-Informationen profitieren und eine größere Anwendungsflexibilität erzielen.

Am Beispiel der Simulation des Magdeburger Straßenbahnbetriebs wird demonstriert, wie online Informationen über reale Prozesse zur Abgleichung oder Initialisierung von Simulationsmodellen eingebunden werden [8,9]. Die erfaßten aktuellen Standortinformationen der Straßenbahnen aktualisieren ständig den Zustand des Simulationsmodells, d. h. es ist immer auf dem aktuellen Stand. Aufwendige Initialisierungen des Modells entfallen.

Der Prototyp besteht aus den Federates zur Simulation, Visualisierung und Online-Daten-Integration und ist in seiner Struktur in Abbildung 4 dargestellt (weitere, zukünftig geplante Komponenten sind in grau hinterlegt). Die aktive oder passive Rolle einer Komponente kann man an der Richtung der Pfeile von bzw. zur Infrastruktur, welche die Richtung des Datenflusses angeben, erkennen.

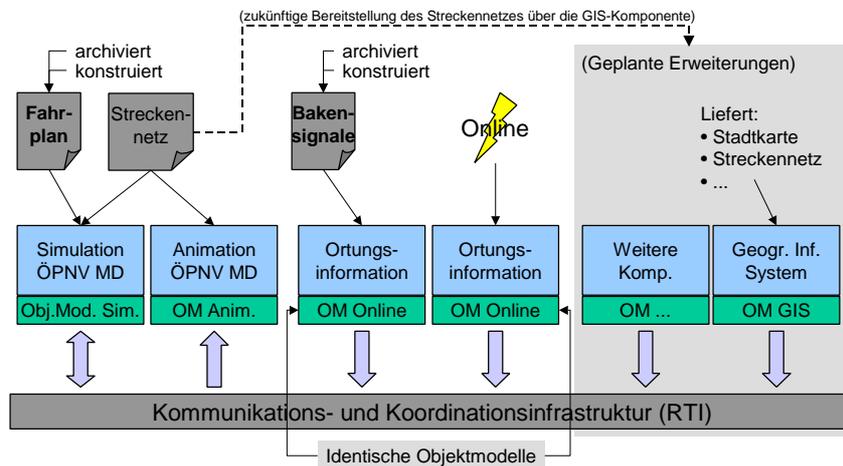


Abbildung 4: Struktur und Komponenten des ÖPNV-Prototypen

Das Simulationsmodell operiert auf der Basis des Soll-Fahrplans (als Offline-Datenquelle) und kommuniziert mit der Online-Komponente, die wahlweise mit dem Echtzeit-Standortsystem der MVB verbunden ist, über welches das Simulationsmodell permanent mit der erfaßten aktuellen Situation abgeglichen werden kann, oder im Offline-Modus unter Verwendung archivierter Daten betrieben wird. Für das Simulationsfederate ist es nicht relevant, ob es Ortungsinformationen von einem Online- oder einem Offline-Federate erhält, oder ob es als reine Simulation ohne Positionsupdates von anderen Federates operiert.

Das Simulationsfederate wurde mit dem Simulationssystem SLX unter Verwendung des SLX-HLA-Interfaces entwickelt [11]. Als Visualisierungsfederate können Skopeo Animation oder Proof Animation verwendet werden. Abbildung 5 zeigt eine Bildschirmkopie der Animation mit Proof Animation.

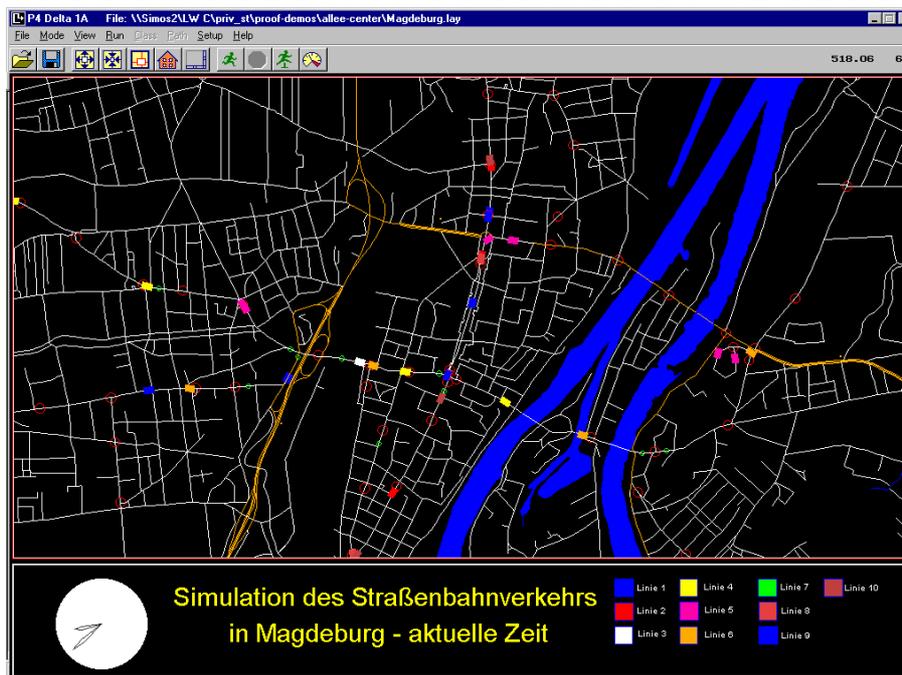


Abbildung 5: Screenshot der Visualisierung durch Proof Animation

## 5. Zusammenfassung

Anhand der an der Universität Magdeburg in Zusammenarbeit mit verschiedenen Partnern entwickelten Prototypen wurde das Anwendungspotential von HLA evaluiert und als praxistauglich befunden (siehe auch [12]). HLA weist gerade auch für die Anwendung im zivilen Bereich ein großes Potential zur Unterstützung von Interoperabilität und Wiederverwendbarkeit von Simulatoren und Simulationsmodellen auf. Für die praktische Anwendung stehen verschiedene Schnittstellen für Simulatoren zur Verfügung, die sich zum Teil in ihrer Anwenderfreundlichkeit und dem Grad der zur Kopplung von Modellen

notwendigen HLA-Kenntnisse erheblich unterscheiden [2,3,4,10]. Für eine echte (durchaus erstrebenswerte und in Zukunft vielleicht sogar durch HLA mögliche) Plug-and-Play Interoperabilität von existierenden Simulationsmodellen reichen die derzeitigen Ansätze noch nicht aus. Der einzige bisher realisierte Ansatz, der die HLA-Fähigkeit eines Simulationssystems gänzlich unter der Oberfläche versteckt, wurde mit Simplex 3 realisiert.

## 6. Literatur

- [1] Henriksen, J.O. 1997. An Introduction to SLX. In Proceedings of the 1997 Winter Simulation Conference, eds. Andradóttir, S., K. Healy, D. Withers, and B. Nelson, pp. 559-566, SCS, Atlanta.
- [2] Straßburger, S. On the HLA-based Coupling of Simulation Tools. In Proceedings of the 1999 European Simulation Multiconference, ed. H. Szczerbicka, pp. 45-51 (Vol. 1). SCS, Warsaw, Poland.
- [3] Straßburger, S., T. Schulze, U. Klein, J.O. Henriksen. 1998. Internet-based Simulation using Off-the-Shelf Simulation Tools and HLA. In Proceedings of the 1998 Winter Simulation Conference, eds. Medeiros, D., E. Watson, J. Carson, and M. Manivannan, pp. 1669-1676. SCS, Washington.
- [4] Lantzsich, G. HLA Interface für Simplex III. Diplomarbeit. Technische Universität Dresden.
- [5] Straßburger, S., T. Schulze, G. Lantzsich. Simplex 3 und SLX - gemeinsam unter HLA. 13. Symposium Simulationstechnik ASIM 99, 21.09.-24.09.1999, Weimar.
- [6] Klein, U., T. Schulze, S. Straßburger, and H.-P. Menzler. 1998. Traffic Simulation Based on the High Level Architecture. In Proceedings of the 1998 Winter Simulation Conference, eds. Medeiros, D., E. Watson, J. Carson, and M. Manivannan, pp. 1095-1103. SCS, Washington.
- [7] Klein, U., T. Schulze, S. Straßburger, and H.-P. Menzler. 1998a. Distributed Traffic Simulation Based on the High Level Architecture. In Proceedings of the Simulation Interoperability Workshop Fall 1998, Orlando.
- [8] Schulze, T., S. Straßburger, U. Klein. 1999. On-line Data Processing in Simulation Models: New Approaches and Possibilities Through HLA. In Proceedings of the 1999 Winter Simulation Conference, pp. 1602-1609. SCS, Phoenix.
- [9] Schulze, T., S. Straßburger, U. Klein. On-Line Data Processing in a Civil Transportation Federation. In: Proceedings of the Simulation Interoperability Workshop, Fall 1999, Orlando.
- [10] Mertins, K., M. Rabe, P. Rieger. Einsatz von Simulations-Referenzmodellen für eine effiziente Erstellung von

Simulationsverbunden auf Basis von HLA. In *Engeli, M. and V. Hrdliczka (Ed.), 12. Symposium Simulationstechnik ASIM 98*. Sept. 15-18, 1998. pp. 299-305.

- [11] Straßburger, S., U. Klein. Integration des Simulators SLX in die High Level Architecture. Tagung Simulation und Visualisierung 1998 Magdeburg. Lorenz, P., Preim, B. (eds.), SCS Europe Publishing House, Delft, Erlangen, Ghent, San Diego. Seiten 32-40.
- [12] Schulze, T., S. Straßburger, U. Klein. Migration of HLA into Civil Domains: Solutions and Prototypes for Transportation Applications. *SIMULATION*, Vol. 73, No. 5, pp 296-303, November 1999.