

Neue Konzepte für die städtische Verkehrssimulation

Thomas Schulze/ Steffen Straßburger

Otto-von-Guericke Universität Magdeburg, Fakultät für Informatik
39106 Magdeburg, Universitätsplatz 2
{tom, strassbu}@isg.cs.uni-magdeburg.de

Ulrich Klein

Fraunhofer Institut für Fabrikbetrieb und –automatisierung (IFF)
39106 Magdeburg, Sandtorstraße 22
uklein@iff.fhg.de

Zusammenfassung

Der Beitrag beschreibt die Anwendung neuer Konzepte auf das Gebiet der städtischen Verkehrssimulation. Diese Konzepte basieren auf der Verwendung der High Level Architecture (HLA) als informationstechnische Basisarchitektur zur Kopplung von Simulationsmodellen und anderen Anwendungen. Die neuen Konzepte beziehen sich auf das kooperative Arbeiten unterschiedlicher verteilter Simulationsmodelle in einer HLA-Federation, die Integration von online-Prozeßdaten zur permanenten Initialisierung von Simulationsmodellen und die Visualisierung von verteilten Simulationen in Netzwerken. Die Verwendung von HLA ermöglicht die Migration der städtischen Verkehrssimulation aus der traditionellen Einzelrechnerumgebung in das Internet. Damit werden neue Anwendungsfelder erschlossen und ein breiterer Nutzerkreis angesprochen. Die Prototypen, anhand derer diese Konzepte erfolgreich implementiert und demonstriert wurden, werden beschrieben.

1 Verkehrssimulation in der Stadt

Entscheidungen und Maßnahmen im Verkehrsbereich einer Stadt haben einen signifikanten Einfluß auf das Gesellschafts- und Wirtschaftssystem Stadt. Es ist zwingend notwendig, die Auswirkungen von vorgesehene Veränderungen an der Verkehrsinfrastruktur oder den verkehrsregelnden Einrichtungen im voraus abzuschätzen. Dies gilt sowohl für den Personen- als auch Güterverkehr. Die Bestimmung der möglichen Auswirkungen erfolgt überwiegend unter der Nutzung von Simulationsmodellen. Mit der Verfügbarkeit von leistungsstarker Hardware und spezieller Simulationssoftware hat sich die Simulation zu einem unverzichtbaren Instrument in der Forschung als auch in der täglichen Praxis entwickelt.

Modelle sind vereinfachte Abbildungen der Realität und stellen die Ausgangspunkte für die Simulation dar. In diesen Modellen sind Strukturen und Komponenten der Realität, ihre Wirkmechanismen und ihre gegenseitigen Einflußnahmen beschrieben. Ein Modell wird nicht die Realität in ihrer Gesamtheit abbilden, sondern nur die wesentliche Sachverhalte werden in das Modell übernommen. Soll z.B. ein neues Steuerprogramm für eine Lichtsignalanlage (LSA) hinsichtlich der Auswirkungen auf die Wartezeiten untersucht werden, so sind Verkehrsdichte und die Fahrzeugtypen im Modell zu berücksichtigen, während z.B. die Farbe der Fahrzeuge keinen Einfluß auf die Wartezeiten hat und daher nicht in das Modell übernommen wird.

2 Monolithische Straßenverkehrssimulation

Simulationsmodelle für den Straßenverkehr werden traditionell als monolithische Modelle entwickelt. Ein monolithisches Modell beinhaltet alle für die Modellierung notwendigen Informationen in einem einzigen Modell. Im Gegensatz dazu besteht ein

verteiltes Simulationsmodell aus mehreren Teilmodellen, wobei jedes Teilmodell einen bestimmten Anteil vom Gesamtmodell beschreibt. Das Zerlegen eines Modells in Teilmodelle kann in Abhängigkeit verschiedener Faktoren sinnvoll sein. Straßenverkehrsmodelle lassen sich z.B. nach geographischen Regionen (für jeden Knoten eines Verkehrsnetzes wird ein Teilmodell verwendet) oder nach den Verkehrsarten (es werden unterschiedliche Teilmodelle für Fahrzeuge, Fußgänger, ÖPNV usw. verwendet) unterteilen.

Mikroskopische Straßenverkehrsmodelle beschreiben den Straßenverkehr mit einem hohem Grad an Detailliertheit. Monolithische mikroskopische Modelle enthalten eine Vielzahl von statischen Komponenten zur Beschreibung der Infrastruktur (Fahrbahnen, Kreuzungen, Lichtsignalanlagen usw.), von dynamischen Komponenten (Fahrzeuge, Fußgänger, Radfahrer usw.) und von Entscheidungsregeln zur Beschreibung der Beziehungen zwischen den dynamischen Komponenten. Durch die Anzahl der beteiligten Komponenten und die Anzahl der Relationen zwischen ihnen lassen sich diese Modelle als large-scale Modelle charakterisieren. Monolithische mikroskopische Simulationsmodelle werden zu groß, zu unhandlich in der Wartung und Anpassung an neue Bedingungen der Realität. Diese Umstände führen zu:

- langen Laufzeiten für die Simulationsläufe,
- langen Entwicklungs- und Testzeiten bei der Verwendung von monolithischen Modellen,
- einem hohen Aufwand bei der Wartung der Modelle bzw. ihrer Anpassung an neue Aufgaben und
- geringer Flexibilität und Wiederverwendbarkeit.

Was kann gegen diese Umstände unternommen werden?

Die Verwendung von leistungsstärkerer Hardware hat nur auf die Laufzeiten der Modelle einen direkten Einfluß. Neue Möglichkeiten ergeben sich durch die Aufteilung von monolithischen Modellen in eine entsprechende Menge von Teilmodellen. Die Reduzierung der Laufzeit ist ein traditionelles Ziel für verteilte Modelle zur Straßenverkehrssimulation [1]. Gegenwärtig zeichnen sich neue Ziele für die Anwendung der verteilten Simulation ab [2]. Diese neuen Zielstellungen werden mit Interoperabilität und Wiederverwendbarkeit umschrieben und adressieren die o.g. Nachteile. Im nächsten Abschnitt wird auf eine aktuelle Architektur zur Umsetzung der verteilten Simulation eingegangen.

3 Die High Level Architecture (HLA)

Die High Level Architecture (HLA) ist eine vom amerikanischen Department of Defense (U.S. DoD) initiierte Architekturform, die sich zu einem IEEE Standard für die verteilte Simulation entwickeln wird [3]. Sie ist ferner bereits durch das DoD und die NATO standardisiert sowie als Facility for Distributed Simulation Systems in die Common Object Request Broker Architecture (CORBA) der Object Management Group (OMG) eingebettet worden. Diese Architektur ist durch folgende Merkmale bestimmt:

- Regeln bestimmen das Verhalten einzelner Simulationsmodelle (Federates) und der Gesamtheit aller Modelle (Federation).
- HLA definiert eine einheitliche Schnittstelle (HLA-Interface) für alle Federates, um in Kontakt mit der Runtime Infrastructure (RTI) zu treten. Die RTI stellt Basis-Koordinations- und Kommunikationsdienste zur Laufzeit bereit. Eine Kommunikation findet lediglich zwischen Federate und RTI und nicht zwischen den Federates selbst statt. Die RTI als zentrale Software-

Koordinierungskomponente kann auf einem beliebigen Netzwerkcomputer lokalisiert sein.

- Eine Federation kann als ein Vertrag zur Durchführung eines Simulationslaufes zwischen den Federates angesehen werden, in dem die Einzelheiten und Objektmodelle der Federates und der Federation festgelegt sind.

Der nächste Abschnitt stellt Simulations- und Animationstools vor, mit denen HLA-Federates erstellt werden können.

4 Werkzeuge zur Simulation mit HLA

Der Ursprung der HLA-Technologie liegt im Bereich der militärischen Simulationsmodelle. In diesem Applikationsfeld erfolgt die Implementierung der Modelle vorwiegend in allgemeinen Programmiersprachen wie C bzw. C++. Diese Simulationsmodelle erfüllen softwareseitig die Bedingungen zum Anschluß an die RTI-Softwarebibliotheken. Zivile Anwendungsbereiche sind durch eine andere Situation geprägt. Die verwendeten Simulationstools entbinden den Nutzer von der Notwendigkeit, seine Modelle auf der Ebene von C bzw. C++ zu implementieren. Die verwendeten Tools sind überwiegend für eine autonome stand-alone Nutzung entwickelt wurden.

Verfügbare Simulations- und Animationstools wurden hinsichtlich ihrer Verwendbarkeit in einer HLA-Architektur untersucht. Im Ergebnis wurden Tools ausgewählt, die dann hinsichtlich ihrer HLA-Kompatibilität erweitert wurden:

- Ein wesentliches Merkmal des Simulationstools SLX (von der Wolverine Software Corporation [4]) ist seine offene Architektur und damit die Möglichkeit zum Aufruf von Funktionen aus einer Windows Softwarebibliothek (DLL). Die Entwicklung der SLX-Version 1.0 erfolgte nicht mit dem Ziel eines Tools zur verteilten Simulation. Durch die Verwendung einer Wrapper-Bibliothek [5] wurde das Standard-SLX erweitert, so daß SLX-Modelle als Federates in einer HLA-Federation operieren können.
- Das Animationstool Skopeo [6] wurde ursprünglich als ein Java-basiertes 2D Werkzeug entwickelt. Auch dieses Tool wurde um die Fähigkeit erweitert, innerhalb einer HLA-Federation Visualisierungsaufgaben zu übernehmen.
- PROOF Animation ist ein weit verbreitetes Werkzeug zur Visualisierung simulierter Prozeßabläufe. Die traditionelle post-run Animation, d.h. die Animation erfolgt zeitlich nach der Simulation, wurde um die Möglichkeit der online Animation erweitert [7]. Die Ausführung des Simulationsmodells und seiner Animation vollzieht sich in diesem Fall zeitlich parallel. Diese Fähigkeit ist zur Visualisierung in verteilten Simulationsumgebungen notwendig. Über eine Windows DLL werden die online-Funktionalitäten aufgerufen. In den Anwendungen zur Verkehrssimulation wurde ein SLX-Rahmenprogramm zur Steuerung der DLL-Aufrufe entwickelt.

Auf der Basis der skizzierten Simulations- und Animationswerkzeuge wurden an der Otto-von-Guericke-Universität Magdeburg mehrere Projekte zur HLA-basierten verteilten Simulation erfolgreich bearbeitet. Im folgenden werden drei Projekte vorgestellt, deren Anwendungsgebiet der städtischen Verkehrssimulation zugeordnet wird. Die Projekte wurden mit unterschiedlichen Zielstellungen entwickelt. Die Homepage des HLA-Competence-Centers Magdeburg bietet eine Übersicht zu weiteren Projekten und Publikationen [8].

5 Einfacher Straßenverkehr

Diese Federation wurde an der Fakultät für Informatik Otto-von-Guericke Universität Magdeburg entwickelt, modelliert ein einfaches Kreuzungsszenario und besteht aus vier unterschiedlichen eigenständigen Federates. Jedes Federate kann unter stand-alone Bedingungen laufen. In der Federation sind drei Simulationsfederates und ein Federate zur Visualisierung vereint. Die Struktur dieser Federation ist Abbildung 1 zu entnehmen.

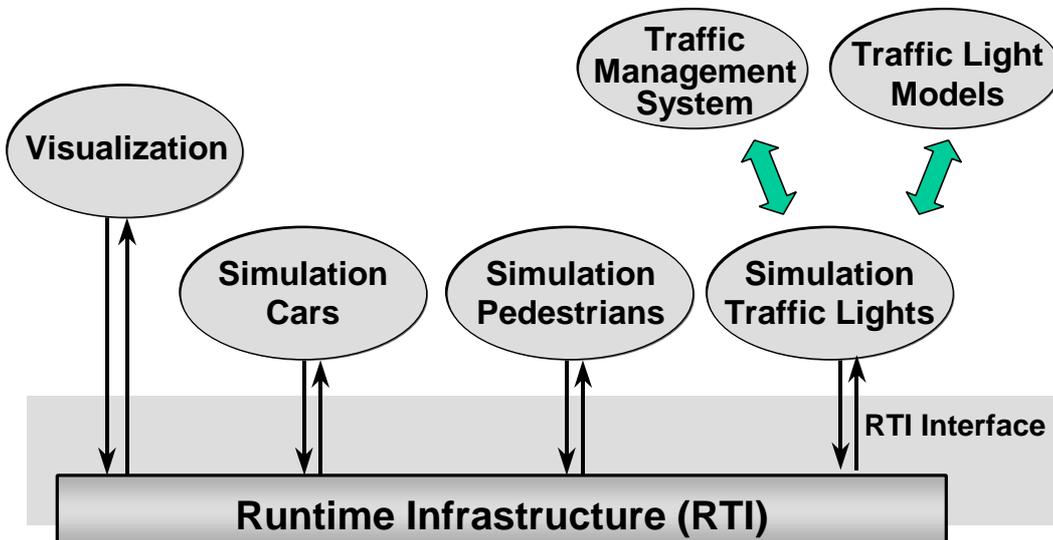


Abbildung 1: Struktur der Federation Einfacher Straßenverkehr

Das Verkehrsmodell wurde in drei eigenständige Simulationsfederates aufgeteilt: *Car Traffic*, *Pedestrian Traffic* und *Traffic Light Control*. Jedes Simulationsfederate wurde mit dem Simulationssystem SLX, unter Verwendung der HLA-SLX-Bibliothek [5] entwickelt. Das Federate Car Traffic simuliert den Fahrzeugstrom auf einer Straße. Eine Lichtsignalanlage (LSA), die durch ein eigenes Federate abgebildet wird, ist in der Straßenmitte aufgestellt. Diese LSA beeinflusst den Auto- und den Fußgängerverkehr. Mit dem Federate Pedestrian Traffic werden Fußgänger simuliert, welche die Straße queren möchten werden. Ein Überqueren der Straße ist nur möglich, wenn die LSA für die Fußgänger ein grünes Licht signalisiert. Im anderen Fall wird der Fußgänger einen Knopf zur Anforderung des grünes Signals drücken. Die Steuerung der LSA wird im Federate Traffic Light Control simuliert. Die Visualisierung der parallelen Prozesse erfolgt im Visualisierungsfederate. Dieses Federate wurde mit dem Tool Skopeo implementiert.

Das zentrale Problem bei der verteilten Simulation ist die Synchronisation der lokalen Simulationuhren auf eine kausalitätserhaltende Weise. HLA erlaubt ein transparentes Zeitmanagement zwischen den unterschiedlichen lokalen Zeiten in den beteiligten Federates. Das Zeitmanagement eines Federates wird durch die folgenden zwei Schalter bestimmt: *time constrained* (es wird auf zeitliche Zwänge anderer Federates reagiert) und *time regulating* (die lokale Zeit anderer Federates wird beeinflusst). Die verwendeten Schalterstellungen sind der Tabelle 1 zu entnehmen.

Tabelle 1 : Federates und die Schalterstellungen zum Zeitmanagement

Federate	Time constrained	Time regulating	Zeitfortschritt
Car Traffic	Ja	Ja	Ereignisorientiert
Pedestrian	Ja	Ja	Ereignisorientiert
Traffic Light Control	Ja	Ja	Ereignisorientiert
Visualisierer	Ja	Ja/Nein	Echtzeitproportional

Die Zielstellungen für die Entwicklung dieses Prototyps lagen schwerpunktmäßig in der Evaluierung von HLA-bezogenen Aufgaben als eine wesentliche Voraussetzung zur Entwicklung praktikabler verteilter Verkehrsmodelle:

- Interoperabilität zwischen ereignisorientierten Simulationsfederates,
- Interoperabilität zwischen Simulations- und Animationsfederates,
- Verwendung des RTI-Zeitmanagements zur Kopplung von ereignisorientierten Simulationsfederates unter Verwendung konservativer Zeitfortschrittsverfahren und
- Verwendung des RTI-Zeitmanagements zur Synchronisation der lokalen Zeiten zwischen ereignisorientierten Simulationsfederates und echtzeitproportionalen Visualisierungsfederates.

Mit diesem Projektes wurde aufgezeigt, daß signifikante informationstechnische Voraussetzungen zur verteilten Verkehrssimulation erfüllt sind und damit neue Konzepte zur städtischen Verkehrssimulation entwickelt und umgesetzt werden können.

6 Verteilte Fahrsimulation

Diese Federation wurde von der Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg (UniMD) gemeinsam mit der Competence Center Informatik GmbH Meppen (CCI) entwickelt. Die beteiligten Partner fügten unabhängig voneinander entwickelte Federates in dieser Federation zusammen. Abbildung 2 illustriert die Struktur dieser Federation. 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.

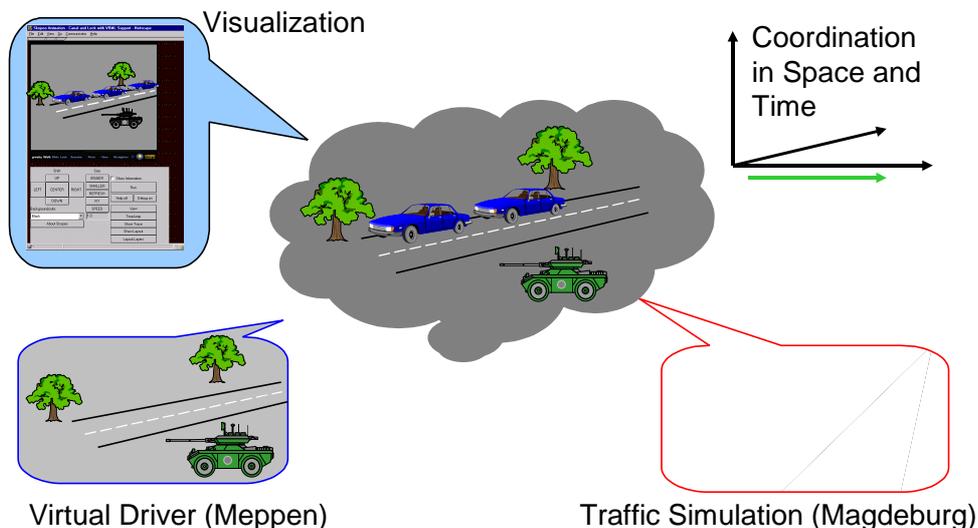


Abbildung 2: Struktur der Federation Verteilte Fahrsimulation

Die beteiligten Federates werden im folgenden kurz charakterisiert:

- Federate **Driving Simulator**: 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++ implementiert und offeriert eine 3D-Visualisierung auf einer Silicon Graphics Maschine. Dieses Federate wurde von CCI entwickelt.
- Federate **Traffic Simulator**: Dieses Federate wurde an der UniMD entwickelt. Es basiert auf einem existierenden ereignisorientierten mikroskopischen Straßenverkehrssimulator, dessen Schwerpunkt auf der Abbildung des psychophysischen Fahrzeugfolgeverhaltens liegt [9]. Dieser Simulator wurde in der Sprache SLX implementiert und läuft auf einem Win32 Standard PC. Mit diesem Federate wird Straßenverkehr nachgebildet. Die Fahrzeuge haben im Zusammenspiel mit dem Federate Driving Simulator auf Aktionen des Fahrers zu reagieren. Im Gegenzug muß der Fahrer im Federate Driving Simulator auf die jeweiligen Fahrzeuge aus dem Straßenverkehr reagieren.
- Federate **Visualization**: Das Beobachterfederate basiert auf dem Visualisierungstool Skopeo.

In der Simulation wurde ein Szenario verwendet, das aus drei Schritten besteht. Die prinzipielle Darstellung der Schritte ist der Abbildung 3 zu entnehmen. Im ersten Schritt startet der menschliche Fahrer die Bewegung des Geländefahrzeuges im Federate Driving Simulator auf einem Gebiet, wo keine Wechselwirkungen mit anderen Verkehrsteilnehmern erfolgen. Dieses Gebiet wird an einer Ausfahrt verlassen, die auf eine Straße mündet. Auf dieser Straße fahren die Fahrzeuge des Federates Traffic Simulator nach festgelegten Verhaltensregeln. Im originären Sinn arbeitet der ereignisorientierte Simulator so schnell wie möglich, d. h. der lokale Zeitfortschritt ist wesentlicher schneller als die reale Zeit. Der lokale Zeitfortschritt muß im Zusammenspiel mit dem in Echtzeit operierenden Federate Driving Simulator synchronisiert werden. HLA stellt hierfür entsprechende Dienste bereit. Beim Einbiegen auf die Straße hat der menschliche Fahrer die simulierten Fahrzeuge des Federate Traffic Simulators zu beachten und eine Lücke im Straßenverkehr zu suchen.

Der zweite Schritt umfaßt das Fahren des Geländefahrzeuges auf der Straße. Der menschliche Fahrer hat auf den simulierten Gegenverkehr und die simulierten Fahrzeuge auf das Geländefahrzeug zu reagieren. Beide Simulationsfederates operieren in einem gemeinsamen Ereignisraum. Sie müssen ihre Zustände in Raum und Zeit koordinieren. Zur Umsetzung dieser Interoperabilität ist HLA ein geeigneter Rahmen. Im dritten Schritt verläßt das Geländefahrzeug die Straße an einer Ausfahrt. Der menschliche Fahrer agiert nun wieder unabhängig von dem simulierten Straßenverkehr.

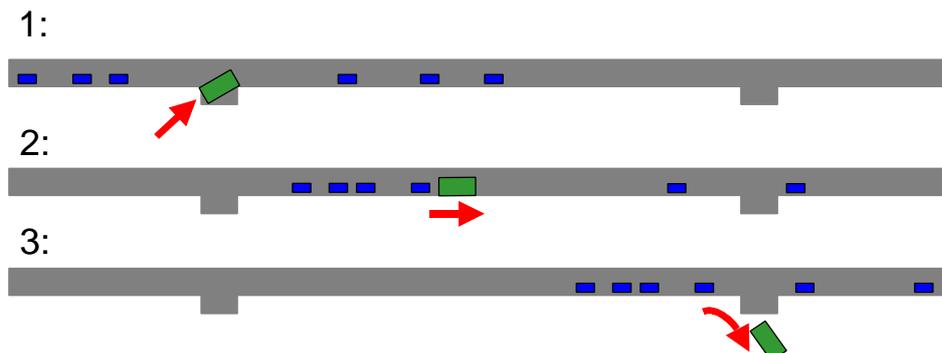


Abbildung 3: Handlungsfolge im Szenario

Die beiden Simulationsfederates mußten ihre alte Funktionalität erweitern. Der Driving Simulator hat zusätzlich die Fahrzeuge auf der Straße zu visualisieren. .
Abbildung 4 zeigt einen Blick von einer Position oberhalb des Geländewagens.

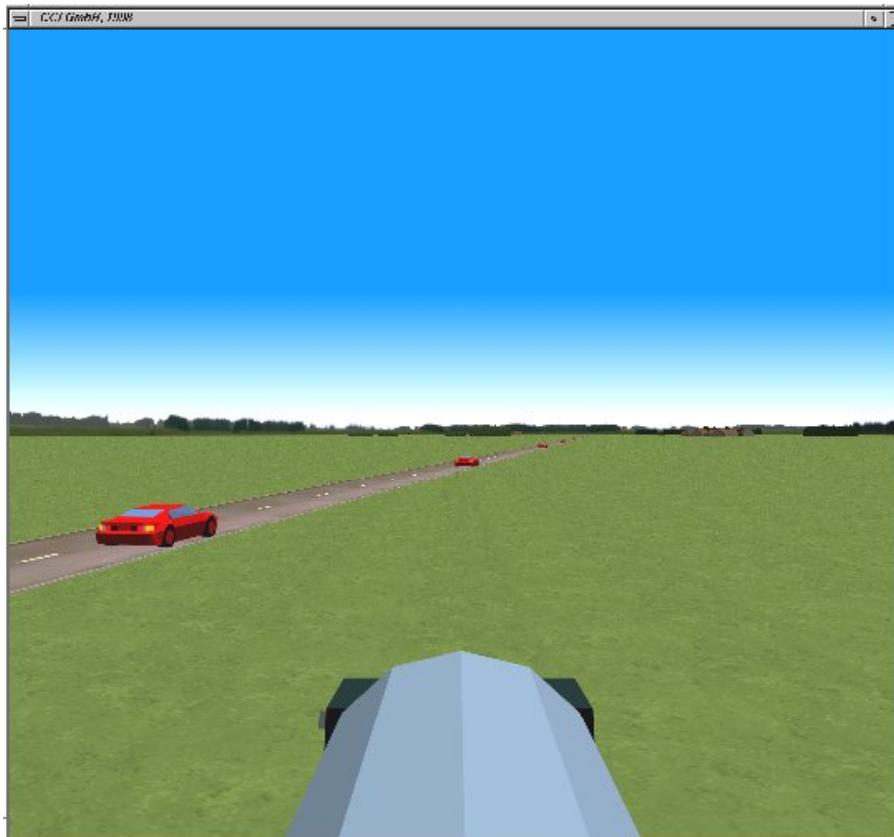


Abbildung 4: Blick vom Geländewagen

Die unterschiedlichen Hardware-Plattformen für die Simualtionsfederates waren während der Simulationsexperimente geographisch getrennt in Meppen und Magdeburg aufgestellt. Als Netzwerkverbindung wurde in diesen Fall eine ISDN-Wählverbindung verwendet. Die Abbildung 5 zeigt die geographische Verteilung der beteiligten Federates.

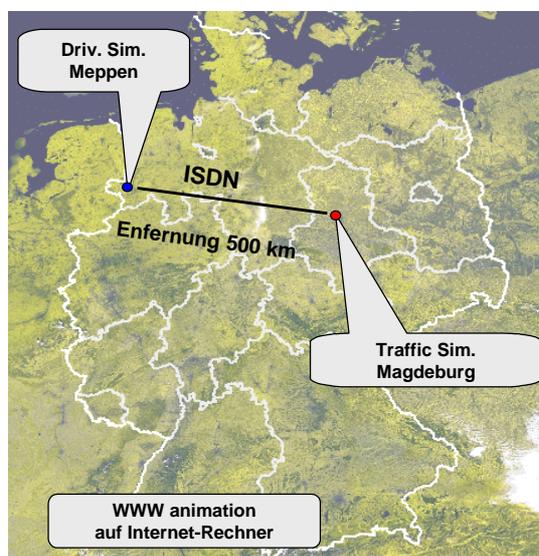


Abbildung 5: Geographische Verteilung der beteiligten Federates

Die Charakteristika des Zeitmanagements in dieser Federation sind in Tabelle 2 zusammengefaßt. Das Federate Driving Simulator agiert *time regulating* und *not time constrained*. Diese Konstellation ist typisch für Federates, die einen Zeitfortschritt in Echtzeit aufweisen und sich nicht durch andere Federates in ihrem Zeitfortschritt beeinflussen. Das Federate Traffic Simulator ist durch *time constrained* und *time regulating* gekennzeichnet. Dieses Federate läßt sich in seinem lokalen Zeitfortschritt durch das Federate Driving Simulator beeinflussen. Die damit verbundene Synchronisation wird von den entsprechenden Diensten des HLA-Zeitmanagements geleistet. Wenn das Federate Driving Simulator nicht der Federation beigetreten ist, so arbeitet das Federate Traffic Simulator „ungebremst“ mit einen ereignisorientierten Zeitfortschritt. Das Federate Visualization agiert als *time constrained* und *not time regulating*. Es übt hinsichtlich des Zeitfortschritts keinen Einfluß auf die anderen Federates aus.

Tabelle 2: Federates und die Schalterstellungen zum Zeitmanagement

Federate	Time constrained	Time regulating	Zeitfortschritt
Driving Simulator	Nein	Ja	Echtzeit
Traffic Simulator	Ja	Ja	Ereignisorientiert
Visualization	Ja	Nein	Echtzeitproportional

Die Federation Verteilte Fahrsimulation zeigte, daß eine unabhängige Entwicklung von Federates und eine Wiederverwendung von existierenden Simulationsmodellen als Federates möglich ist. Die folgenden Aspekte charakterisieren den erfolgreichen Versuch zur Verwendung von HLA als ein Rahmen zur Erreichung des Zieles der Interoperabilität zwischen Simulationsmodellen:

- Interoperabilität zwischen unterschiedlichen Zeitfortschrittmethoden (Echtzeit vs. ereignisorientierter Zeit , ereignisorientierte Zeit vs. proportionaler Echtzeit),
- Interoperabilität zwischen heterogenen Hardwareplattformen (SGI vs. Windows PC vs. plattformunabhängige Java Applikationen),
- Interoperabilität zwischen den Implementationssprachen (C++ vs. SLX vs. Java),
- Interoperabilität zwischen den verwendeten Netzwerken (LAN vs. ISDN-Link vs. Internet).

7 Online-Verarbeitung von Standortinformationen

Beim Einsatz von Simulationsmodellen für Planungsaufgaben werden die benötigten Eingabedaten überwiegend in Datenbanken oder anderen Eingabedateien bereitgestellt. Zur Simulation des Ablaufes an einer zu planenden Straßenkreuzung mit LSA sind die entwickelten Steuerpläne für die LSA, die Geometrie- und Topologiedaten sowie die Kennzahlen der zu erwarten Verkehrsströme in Dateien abgelegt. Man spricht in diesem Zusammenhang auch von der offline Bereitstellung von Eingabedaten. Werden dagegen Simulationsmodelle zur Unterstützung von Steuerungsaufgaben oder zu Trainingszwecken eingesetzt, so sind Teile der benötigten Daten aus den realen Prozessen möglichst online bereitzustellen und zu verarbeiten. In bestimmten Störungssituationen, wie z.B. bei Unfällen mit Straßenbahnen, sind geeignete Maßnahmen einzuleiten. Simulationsmodelle, die mit den aktuellen Standortinformation aller Straßenbahnen initialisiert sind, unterstützen einen Dispatcher bei der notwendigen Entscheidungsfindung. Dies umgeht auch die Problematik des Einschwingverhaltens

bei der Simulation, da sofort auf einem gültigen Systemzustand gearbeitet werden kann. Die aktuellen Standortinformationen sind online zu verarbeiten.

Eine andere Bedeutung für die online-Verarbeitung von aktuellen Prozeßdaten ergibt sich für die Visualisierung realer Prozesse. Ein Visualisierungsmodell ist in der Lage, in periodischen Abständen die aktuellen Informationen zu empfangen und bis zum nächsten Empfang der Daten die mögliche Positionen zu simulieren. Dabei wird gegebenenfalls das Animationsmodell wieder der Realität angepaßt.

Traditionell werden die online-Daten direkt mit dem Simulations- oder Animationsmodell verbunden. Ein neuer Ansatz ist jedoch die Bildung von Komponenten und die Integration von Echtzeit-/Online-Komponenten in die als Interoperabilitätsplattform genutzte High Level Architecture (HLA). Die im folgenden beschriebenen Komponenten lassen sich in Simulations-, Animations- und Online-Komponenten differenzieren. Auch die Verwendung von Offline-Datenkomponenten (d.h. solchen, die keine direkte Verbindung zu einem externen datengenerierenden Prozeß besitzen) wie z.B. Datenbanken sind mit diesem Konzept möglich.

Die Verarbeitung von online Standortinformationen für den städtischen Verkehrsbereich wird für den öffentlichen Personennahverkehr (ÖPNV) aufgezeigt. Der vorgestellte Prototyp bildet den ÖPNV auf der Basis des Straßenbahnverkehrs in der Landeshauptstadt Magdeburg ab. In Zusammenarbeit mit der Magdeburger Verkehrsbetriebe GmbH (MVB) stehen die Fahrplandaten sowie ein Onlinezugang zum Echtzeit-Fahrzeugortungssystem der MVB zur Verfügung.

7.1 Die ÖPNV-Federation

Der Prototyp besteht aus den Funktionen Simulation, Animation und Visualisierung sowie Online-Daten und ist in seiner Struktur in Abbildung 6 dargestellt (weitere, zukünftig geplante Komponenten sind grau hinterlegt). Die Kopplung der Komponenten erfolgt auf der Basis der High Level Architecture (HLA). 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.

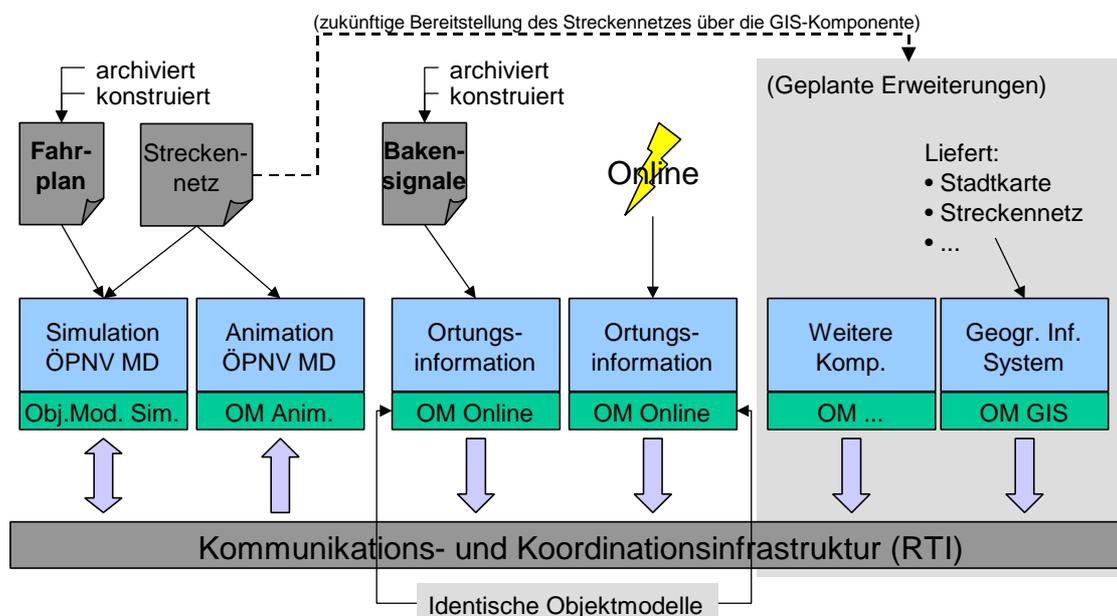


Abbildung 6: Struktur der ÖPNV-Federation

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.

Die beteiligten Federates verfügen in der hier beschriebenen Federationzusammensetzung über unterschiedliche Zeitmanagementmerkmale. Das Animationsfederate richtet sich als passive Komponente nach dem Zeitfortschritt anderer Federates und besitzt daher die Zeitmanagementcharakteristik *time constrained, not time regulating*. Das Simulationsfederate als aktive Komponente reguliert den Zeitfortschritt z.B. des Animationsfederates und ist daher *time regulating*; da das Simulationsfederate auch auf externe kausalgeordnete Ereignisse, wie sie z.B. aktuelle Standortmeldungen darstellen, reagiert, ist es außerdem *time constrained*, also ebenfalls von anderen zeitregulierenden Federates abhängig. Das Online-Datenfederate ist eine aktive, von anderen Komponenten zeitlich unabhängige Komponente, weshalb sie die Merkmale *time regulating, not time constrained* besitzt. Einen Überblick zum Zeitmanagement der Federates gibt Abbildung 7.

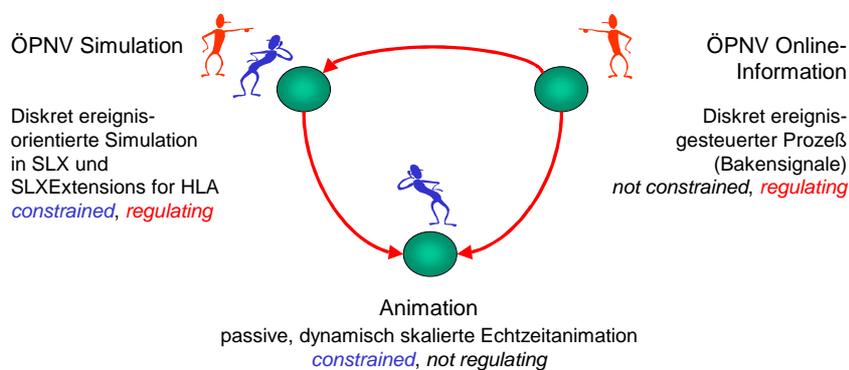


Abbildung 7: Überblick zum Zeitmanagement

7.2 Das ÖPNV-Simulationsfederate

Das Simulationsfederate wurde mit dem Simulationssystem SLX unter Verwendung der *SLX Extensions for HLA* entwickelt.

Das Objektmodell des Simulationsfederates umfaßt das Streckennetz einschließlich aller Haltestellen, des Fahrplans, der Positionen der Baken zur Erfassung der aktuellen Standorte, der Depots und der Bahnen selbst. Die Straßenbahnen sind als aktive Objekte modelliert und bewegen sich auf dem Streckennetz entsprechend ihres Fahrplanes, wobei ein 24-stündiger Bahnbetrieb simuliert wird.

In der derzeitigen Implementierung wird nach Betritt des Online-Datenfederate zur Federation bei jeder empfangenen Standortinformation der Zustand des Simulationsmodells dem realen Zustand angepaßt, d.h. die Straßenbahnen werden, unabhängig von dem bisherigen simulierten Zustand, an die aktuelle Bakenposition gesetzt. Der Zustand im Simulationsmodell wird somit von Simulationsmodell selbst als auch von den externen Online-Daten verändert. Aus diesem Umstand ergeben sich Forderungen an die interne zeitliche Steuerung des Simulators, welcher über spezielle Rescheduling-Fähigkeiten verfügen muß, um geplante Ereignisse zum Zeitpunkt t aus der Liste der geplanten Ereignisse zu entfernen und sie zum Zeitpunkt t in die Liste der aktuellen Ereignisse einzuordnen. Andere Vorgehensweisen des Abgleichs zwischen simulierten und gemeldeten Standorten sind, abhängig von der jeweiligen Anwendung, vorgesehen.

7.3 Das ÖPNV-Visualisierungsfederate

Das Visualisierungsfederate existiert in zwei Versionen, die jeweils mit unterschiedlichen Animationstools entwickelt wurden. Zum einen wurde eine Version mit dem java-basierten plattformunabhängigen Visualisierungstool Skopeo realisiert und zum anderen ist eine Version unter Verwendung des Animationstools PROOF Animation entwickelt worden.

Im Rahmen der ÖPNV-Federation wurde für beide Versionen mit Hilfe eines Geographischen Informationssystems (GIS) eine Vektorkarte der Stadt Magdeburg auf Basis des aktuellen Datenbestandes des Amtlichen Topographisch-Kartographischen Informationssystems (ATKIS) erzeugt, welche für die Bewegungspfadgenerierung und Visualisierung genutzt wurde. Abbildung 8 zeigt eine Momentaufnahme der PROOF-Animation; angezeigt wird die Gesamtansicht, auf der die animierten Straßenbahnen als rote Punkte animiert werden.

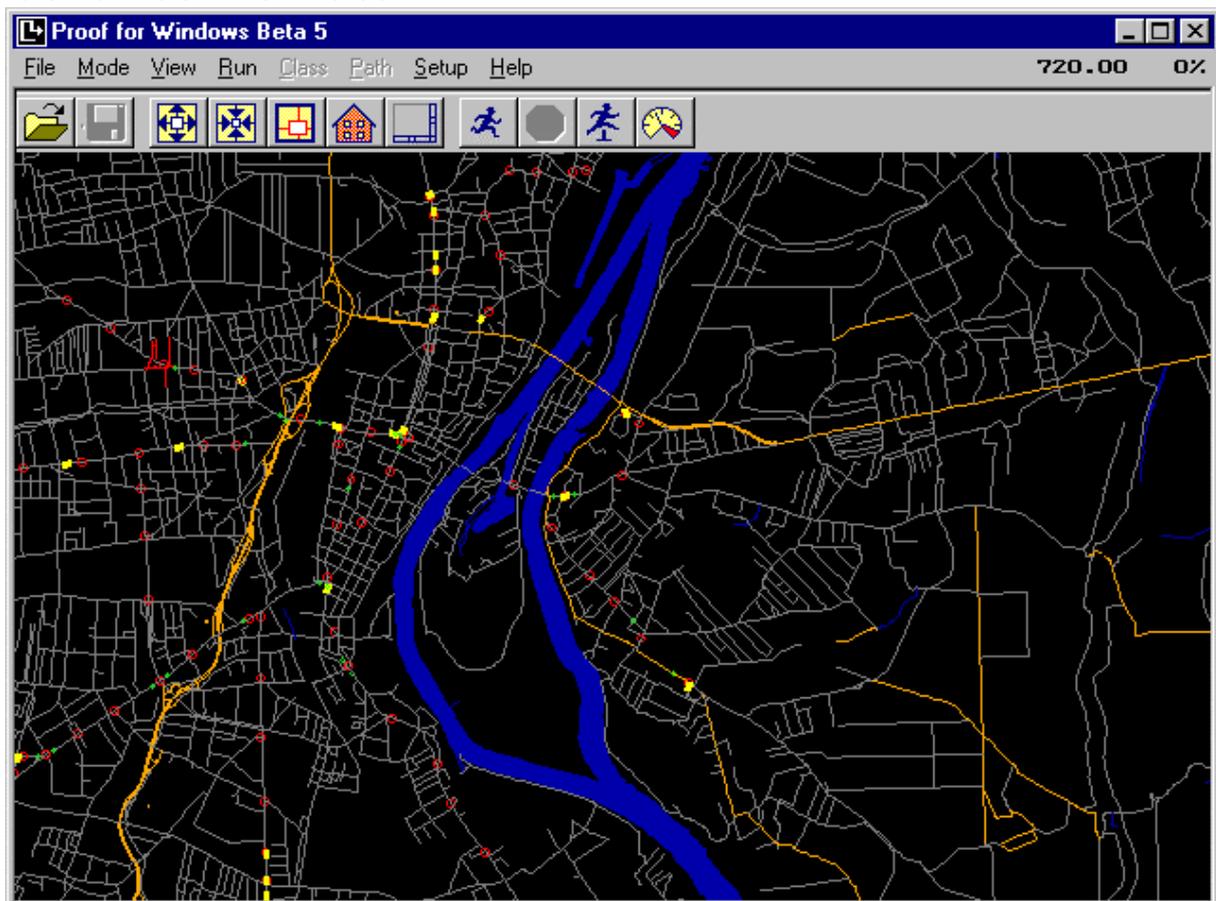


Abbildung 8: Animationsausgabe von PROOF Animation

7.4 Das ÖPNV-Online-Federate für Fahrzeugortungsdaten

Das Online-Federate liest über dessen Empfangsprozess die Online-Daten und stellt sie der Federation analog dem Aufbau in Abbildung 9 zur Verfügung.

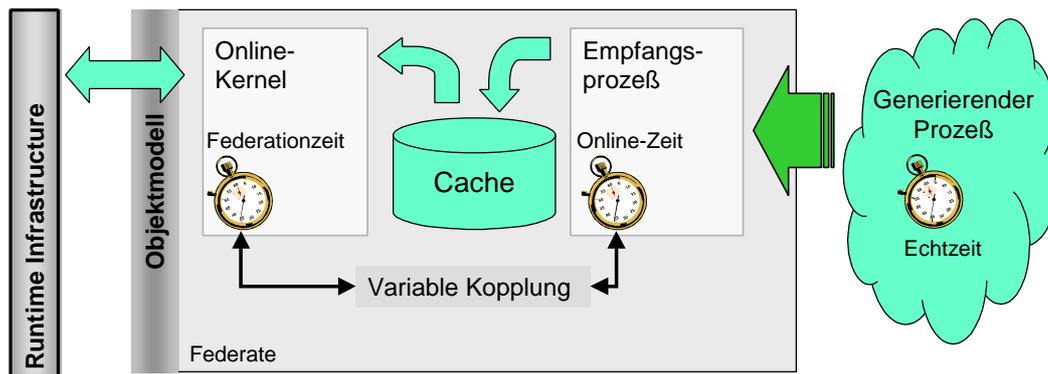


Abbildung 9: Funktionale Übersicht zur Online-Komponente

Eine Online-Komponente hat die Aufgabe, die Online-Daten in geeigneter Form der Federation zur Verfügung zu stellen. Je nach Datenbeschaffenheit der in Echtzeit bereitzustellenden Daten bietet sich dabei entweder eine Modellierung als Objekte oder als Wechselwirkungen (*Interactions*) an. Das erforderliche Objektmodell läßt sich hieraus und unter Berücksichtigung von Modellierungsalternativen und sonstiger Randbedingungen der Federation ableiten.

Um der Natur der empfangenden Komponenten sowie deren Zeitfortschrittsmechanismen so weit wie möglich gerecht werden zu können, ist ein flexibler Aufbau notwendig, welcher die zeitliche Entkopplung der empfangenen Echtzeitdaten (Empfangsprozeß) und dem federationinternen Zeitfortschritt erlaubt. Je nach einzunehmender Rolle der Echtzeitkomponente ist diese flexibel konfigurierbar.

Die Konfigurierbarkeit wird im wesentlichen durch die Art der Kopplung zwischen der Online- und Federationzeit sowie der Konfiguration des Zeitmanagements unter HLA bewirkt:

- Die variable Kopplung der Online-Zeit (die Zeitstempel der eintreffenden Online-Informationen einschließlich zu berücksichtigender Abweichungen und Übertragungslatenzen) mit der Federationzeit (die Zeit, welche die Federation aus der Sicht des Federates innehat) reicht von starren bis hin zu losen Kopplungen. Die starre Kopplung charakterisiert einen Zustand, bei dem die eingehenden Online-Informationen direkt an die Federation weitergereicht werden. In diesem Fall ist die Online-Zeit mit der Federationzeit in Übereinstimmung. Eine lose Kopplung liegt vor, wenn die eingehenden Online-Daten gespeichert oder manipuliert werden müssen. Beispiele hierfür sind der Ausgleich bei unregelmäßigen Zeitfortschritt in der Federation, latenzausgleichende Zeitstempelkorrekturen, Verzögerungen oder Werteinterpolationen.
- Das Zeitmanagement unter HLA fordert für jedes Federate die Festlegung von zwei Attributen. Aus der Kombination der Attribute *time-constrained* und *time-regulating* ergeben sich unterschiedliche Eigenschaften und damit Anwendungsfälle für eine Online-Komponente. Eine unabhängige Echtzeit-Online-Komponente trüge die Charakteristik *not time constrained, not time regulating*; eine Unterordnung unter den Federationzeitfortschritt (bei loser zeitlicher Kopplung der Onlinedaten) wäre durch *time constrained, not time regulating* erreichbar. Für eine Taktgeberfunktion wäre *not time constrained, time regulating* verwendbar, und zur Abbildung komplexerer zeitlicher Abhängigkeiten innerhalb einer Federation ließe sich *time constrained, time regulating*, verwenden.

Die gemessenen Ortungsinformationen nehmen dabei den in Abbildung 10 beschriebenen Verlauf:

- (1) Die Straßenbahnen "identifizieren" sich bei den Baken und diese senden über ein internes Netz ein Informationstelegramm an den Standortinformationsrechner der MVB. Das Telegramm enthält Informationen über Zeitpunkt, Bakennummer und Straßenbahnidentifikation.
- (2) Der Standortinformationsrechner sendet über ein Modem diese Informationen zu einem Rechner an der Universität Magdeburg.
- (3) Um Standort- und Plattformunabhängigkeit zu erlangen, wurde für den weiteren Kommunikationsweg ein TCP/IP-Port als Kommunikationsmedium gewählt, der von dem Empfangsprozess des Online-Federate abgefragt wird. Das Online-Federate stellt dann die empfangenen Daten der Federation zur Verfügung.

Das Federate wurde ebenfalls mit dem Simulationssystem SLX und den *SLX Extensions for HLA* entwickelt.

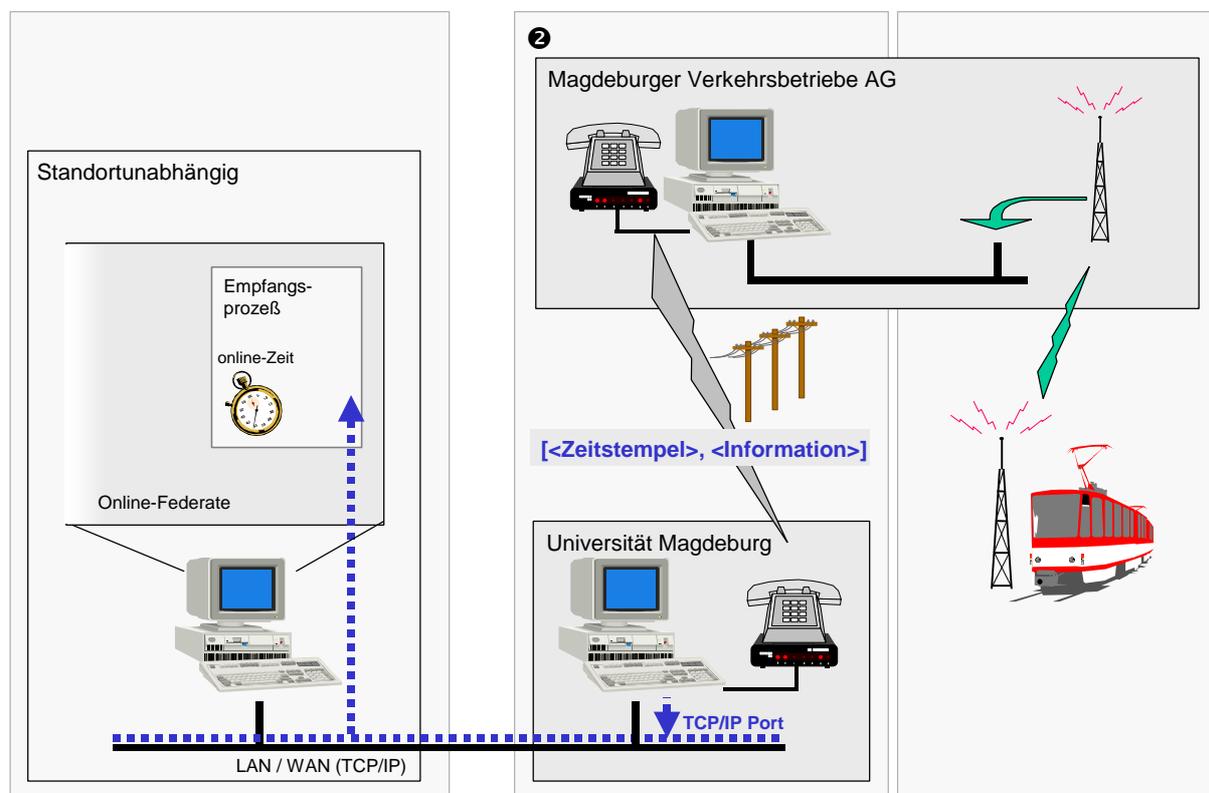


Abbildung 10: Kommunikationsweg für die Online-Informationen

7.5 Das ÖPNV-Offline-Federate für Fahrzeugortungsdaten

Wie in Abschnitt 7.1 beschrieben, kann als Informationsquelle für die aktuellen Bahnstandorte in dem Online-Federate auch eine Datei verwendet werden. In dieser Betriebsart (Offline-Modus) arbeitet die Online-Komponente als „Offline-Federate“ und die Fahrzeugstandortinformationen werden aus einem Archiv in Dateiform bezogen.

Diese Informationen werden für as-fast-as-possible konfigurierte Federationen so schnell ausgelesen, wie es der Zeitfortschritt in der Federation erlaubt (daher hat das „Offline-Federate“ die Zeitmanagementmerkmale *time constrained, time regulating*).

Für einen echtzeitskalierten Zeitfortschritt, wie es Anwendungen zur Visualisierung oder zum Training erfordern, wird eine den Zeitstempeln der Standortnachrichten entsprechende zeitliche Verzögerung berücksichtigt. Im letzteren Fall kann das „Offline-Federate“ zur Herstellung einer Echtzeitproportionalität eine Taktfunktion übernehmen, falls dies nicht bereits anderweitig innerhalb der Federation gewährleistet wird.

Ebenso wie die Online-Variante wurde dieses Federate mit dem Simulationssystem SLX und den *SLX Extensions for HLA* entwickelt.

7.6 Einsatzvarianten

Durch eine unterschiedliche Zusammensetzung der Komponenten kann der Prototyp zu unterschiedlichen Aufgaben eingesetzt werden (Kompositionsprinzip). Für klassische Planungszwecke werden z.B. die Komponenten Simulation, Offline-Daten und Animation verbunden. Eine Online-Visualisierung des aktuellen Zustandes unter Echtzeitbedingungen, als Basis für innovative Fahrgastinformationssysteme, benötigt die Komponenten Animation und Online-Daten. Die Anwendung für Steuerungszwecke und zur operativen Auswertung von Handlungsalternativen bedingt die Komponenten Simulation, Offline-Daten, Online-Daten und Animation. Zum Dispatchertraining empfiehlt sich die Kombination der Komponenten Simulation, Offline-Daten und interaktive Animation (Tabelle 3).

Tabelle 3: Einsatzvarianten für die Komposition der ÖPNV-Federation

Simulation	Animation	Online	Anwendungsbeispiele
x	x		Klassisches Planungstool
	x	x	Online-Visualisierung, z.B. in einem Fahrgastinformationssystem
x		x	Experimente unter Verwendung des aktuellen Systemzustandes
x	x	x	Dispatchertraining

Die Anwendung des Kompositionsprinzips, d.h. der Zusammenbau aus bestehenden, z.T. heterogenen Komponenten, ermöglicht eine flexible Gestaltung von Tools zur Modellierung und Simulation, zum Training und zum Management von geplanten oder real existierenden Prozessen. Mit der HLA ist eine Integrationsplattform verfügbar, die es ermöglicht, diese Komponenten flexibel, den gestellten Anforderungen entsprechend, zu verbinden. Die Integration von Echtzeit-/Online-Daten realer Prozesse für den Bereich des ÖPNV wurden in einem Prototyp aufgezeigt.

Diese Vorgehensweise ist auch auf anderen Applikationsgebiete übertragbar. Hierzu zählen u.a. das Flottenmanagement für Logistikanbieter, die Fertigungsteuerung in produzierenden Unternehmen und Trainingssimulatoren. Voraussetzung ist die Verfügbarkeit von HLA-fähigen Komponenten, deren Zahl stetig wächst.

8 Zusammenfassung und Ausblick

Der Beitrag beschreibt die Anwendung zweier neuer Konzepte, der verteilten Simulation und der Einbettung in das Internet, auf das Gebiet der städtischen Verkehrssimulation. Die Prototypen, anhand derer diese Konzepte erfolgreich implementiert und demonstriert werden konnten, sind beschrieben. Die hier aufgezeigte Entwicklung und neue Ausrichtung auf Eigenschaften wie Interoperabilität und Wiederverwendbarkeit stehen erst am Anfang. Es steht zu erwarten, daß mit die Leistungsfähigkeit der HLA neue Anwendungsfelder erschließen und die Integration von Softwarekomponenten vorantreiben wird.

9 Literatur

[1] Mabry, Susan L., Jean-Luc Gaudiot. Distributed parallel object-oriented environment for traffic simulation (POETS). In: *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, S. Manivannan, D. A. Sodowski, and A. F. Seila, pages 1093-1100.

[2] Klein, U. und S. Straßburger. Die High Level Architecture: Anforderungen an interoperable und wiederverwendbare Simulationen am Beispiel von Verkehrs- und Infrastruktursimulationen. In *Simulationstechnik, Tagungsband 11. Symposium in Dortmund*, November 1997, ed. A. Kuhn und S. Wenzel. Seiten 529-534, Vieweg 1997.

[3] Defense Modeling and Simulation Office (DMSO). 1998. The High Level Architecture. Homepage. Online available at : <http://hla.dmsomil>.

[4] Schulze, Th. und J. Henriksen. Simulation Needs SLX. Handbuch zum Simulationssystem SLX. Otto-von-Guericke Universität Magdeburg, Fakultät für Informatik 1998.

[5] Straßburger, S. und U. Klein. Integration des Simulators SLX in die High Level Architecture. In *Proc. Simulation und Visualisierung' 98, Magdeburg*. Eds. P. Lorenz und B. Preim, März 5-6. 1998, SCS Europe, Seiten 32-40.

[6] Ritter, K. C., U. Klein, S. Straßburger and M. Diessner. Web-basierte Animation verteilter Simulationen auf der Basis der High Level Architecture (HLA) In *Proc. Simulation und Visualisierung' 98, Magdeburg*. Eds. P. Lorenz und B. Preim, März 5-6. 1998, SCS Europe, Seiten 41-52.

[7] Henriksen, J. GENERAL-PURPOSE CONCURRENT AND POST-PROCESSED ANIMATION WITH PROOF™. In: *Proceedings of the 1999 Winter Simulation Conference*, (Seitenangabe noch nicht verfügbar)

[8] HLA in Magdeburg. Homepage. Online available at : <http://isgsim.cs.uni-magdeburg.de/hla/>

[9] Schulze, Th. und Th. Fliess. Entwicklung eines SLX-basierten mikroskopischen Straßenverkehrssimulators. In *Simulationstechnik, Tagungsband 11. Symposium in Dortmund*, November 1997, ed. A. Kuhn und S. Wenzel. Seiten 345-350, Vieweg 1997.