

Simplex 3 und SLX – gemeinsam unter HLA

Steffen Straßburger / Thomas Schulze

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

Gunter Lantzsich

Universität Passau, Lehrstuhl für OR und Systemtheorie
94032 Passau, Innstraße 33
lantzsich@lisa.fmi.uni-passau.de

Zusammenfassung

Der Beitrag zeigt anhand der Simulationssysteme Simplex 3 und SLX, daß es mittels der High Level Architecture (HLA) möglich ist, grundverschiedene Simulationssysteme miteinander zu koppeln. Durch die Integration von HLA-Schnittstellen in bewährte und von Modellentwicklern akzeptierte Simulationssysteme erhält HLA Einzug in den zivilen Bereich. Die für das Simulationssystem Simplex entwickelte Vorgehensweise zur Anbindung von Simulatoren an eine HLA-Federation zeichnet sich dadurch aus, daß ein Paradigmenwechsel bei der Erstellung von HLA-kompatiblen Modellen nicht notwendig ist. Dadurch wird die Verbreitung und Akzeptanz von HLA wesentlich unterstützt.

1 Motivation – Einsatz von HLA im zivilen Sektor

Die HLA bietet einen Ansatz, die Forderungen nach Wiederverwendbarkeit und Interoperabilität von Simulationsmodellen umzusetzen und die Schwierigkeiten mit monolithischen Modellen bei Änderungen der Funktionalität oder Konnektivität zu umgehen.

HLA definiert eine Architektur zur verteilten Simulation, die es ermöglicht, verschiedenartige (Simulations-)systeme miteinander zu koppeln [1]. Im Gegensatz zur klassischen parallelen Simulation stellt HLA hierbei nicht den Geschwindigkeitsgewinn durch Parallelisierung der Modellausführung in den Vordergrund, sondern die Möglichkeit der Kopplung an sich.

Betrachtet man die Geschichte der Parallelsimulation, so zeigt sich, daß sie meist nur in sehr spezialisierten Fällen Anwendung findet. Eine breite Wirkung hat diese Technologie nicht erreicht. Dies liegt zum einem darin begründet, daß vom Modellentwickler ein Paradigmenwechsel¹ gefordert wird. Die Entwicklung der Modelle erfolgt häufig auf dem Niveau von Programmiersprachen, die um Funktionalitäten zur parallelen Simulation erweitert wurden. Weitverbreitete existierende kommerzielle Simulationssysteme verfügen nicht über die benötigten Features zur Umsetzung physischer Parallelität bei der Ausführung der Simulation. Für eine breite Akzeptanz sind dies ungünstige Faktoren. Um eben jene Akzeptanz zu erreichen, muß daher für eine Technologie wie HLA gefordert werden, daß der Prozeß der Modellerstellung in einer dem Modellierer vertrauten Umgebung stattfindet. Ob der Modellierer zwei klassische Teilmodelle schreibt, die innerhalb eines Simulators interagieren (als ein klassisches

¹ Unter Modellierungsparadigma sei hier die Sichtweise auf das Modell und die Art und Weise der Modellentwicklung verstanden.

monolithisches Simulationsmodell), oder ob die zwei Teilmodelle über HLA gekoppelt werden, sollte auf den Modellierungsprozeß keinen Einfluß haben. Anhand der Simulatoren Simplex und SLX wurde untersucht, inwieweit sich dieses Ideal verwirklichen läßt. Eine Einschätzung der Chancen von HLA im zivilen Bereich wird daraus abgeleitet.

2 Typen von HLA-Anbindungen

Für die Simulatoren Simplex 3 und SLX wurden HLA-Schnittstellen entwickelt, die auf zwei völlig unterschiedlichen Ansätzen basieren [2, 3]. Für das Simulationssystem Simplex wurde das oben erwähnte Konzept der Transparenz aus Nutzersicht umgesetzt: die Modellbeschreibung eines Simplex-Modells in der Model Definition Language von Simplex ist unabhängig von der Tatsache, ob das Modell als Federate im Sinne von HLA verwendet wird oder ob es ein Teilmodell innerhalb von Simplex ist. Im Falle, daß das Modell als Federate agieren soll, sind vom Nutzer lediglich spezielle Mapping-Dateien zu erstellen, die angeben, wie die Abbildung zwischen internen Simplexdaten und externen HLA-Daten erfolgen soll. Dies erfolgt in Analogie zur Verschaltung von Simplex-Variablen intern über eine High Level Component. Zur Realisierung dieses Konzepts wurde das Laufzeitsystem von Simplex erweitert, so daß es sowohl klassische Simplexmodelle als auch Simplex-Federates ausführen kann. Da bei dieser Form der HLA-Anbindung Aufrufe an die HLA-Basissoftware automatisch bzw. implizit erfolgen, wird sie als „implizite Anbindung“ bezeichnet.

Die HLA-Schnittstelle für SLX verfolgt ein vollständig anderes Konzept: dem Nutzer werden Funktionen bereitgestellt, die denen der HLA-Interface Spezifikation entsprechen; jedoch für den Simulator aufbereitet sind, indem sie den Datentypen und Konventionen von SLX entsprechen. In einem SLX-Simulationsmodell sind dann, um die HLA-basierte Kopplung zu erreichen, explizit spezielle Funktionsaufrufe einzufügen. Daher trägt diese Form der Anbindung die Bezeichnung „explizite Anbindung“.

Beide Ansätze sind durch Vor- und Nachteile geprägt. Die folgende Tabelle faßt diese zusammen und vergleicht sie mit der direkten Erstellung von Federates in Programmiersprachen, wie es im militärischen Simulationssektor üblich ist.

Kriterium \ Eingesetztes System	C++, Java	Simplex	SLX
Gesamtaufwand für die Erstellung eines Federates	☹☹	☺☺	☺
Komfort der Modellerstellung, Nutzerfreundlichkeit	☹☹	☺☺	☺☺
Wiederverwendbarkeit bestehender (monolithischer) Programme /Modelle	☹☹	☺☺	☺
Wiederverwendbarkeit bestehender HLA-kompatibler Programme /Modelle	☺	☺☺	☺☺
Flexibilität des HLA-Interfaces	☺☺	☺	☺
„Learnig-Curve“ bei der Entwicklung von Federates (wieviel muß der Entwickler hinzulernen)	☹☹	☺	☺
Eignung zum Erlernen von HLA	☺☺	☹	☺☺

Tabelle 1: Vergleich der verschiedenen Ansätze zur Erstellung von HLA-Federates

Aus der Tabelle ist erkennbar, daß unter dem Gesichtspunkt einer möglichst transparenten HLA-Anbindung die implizite Anbindung (der Simplex-Ansatz) zu favorisieren ist. Bei der Modellerstellung unter Simplex muß nicht (oder nur gering, z.B.

aus Performancegründen) darauf geachtet werden, ob ein HLA-kompatibles Modell oder ein Simplex-Teilmodell erstellt wird. Daher ist bei der impliziten Anbindung das oben aufgestellte Ziel, daß HLA beim Modellentwickler keinen Paradigmenwechsel erfordern darf, gewährleistet.

Die explizite Anbindung, die beim Simulationssystem SLX angewendet wurde, wird diesem Ziel in seiner ursprünglichen Form noch nicht vollständig gerecht. In einem HLA-konformen SLX-Modell sind für Daten (d.h. Attribute von Objekten), die im HLA-Sinne „public“ sind, Funktionsaufrufe an das HLA-Interface einzubauen. Zusätzlich ist eine separate Synchronisationsroutine zum Modell hinzuzufügen, welche die zeitliche Synchronisation mit anderen Federates übernimmt. Diese Routine ist jedoch unabhängig vom zu entwickelnden Modell.

Zusammenfassend läßt sich sagen, daß auch die Lösung für SLX keinen Paradigmenwechsel bei der Modellerstellung verlangt. Es ist jedoch notwendig, die HLA-Schnittstelle in das Modell zu verlagern und somit das Modell zu erweitern. Eine Unabhängigkeit des Modells gegenüber der Tatsache, ob es ein Teilmodell innerhalb eines komplexeren SLX-Modells oder ein HLA-Federate ist, wie dies bei Simplex der Fall ist, besteht nicht. Weitere Vereinfachungen der Handhabung von HLA und ein weiteres „Verstecken“ der HLA-Funktionalität ist jedoch durch das Statement-Konzept von SLX möglich. Das Statement-Konzept erlaubt die Erweiterung des Sprachumfangs von SLX, so daß dem Nutzer beispielsweise vorgefertigte HLA-Objekte in Abhängigkeit eines speziellen Simulation Object Models bereitgestellt werden können.

Kritisch ist anzumerken, daß sich bei steigendem Automatisierungsgrad des HLA-Interfaces jedoch auch ein Tradeoff bezüglich der Flexibilität feststellen läßt. Die größte Flexibilität liegt verständlicherweise bei der Entwicklung eines Modells auf Programmiersprachenniveau. Es gilt der Grundsatz, daß man kann alles tun kann, man sich jedoch um alles selbst kümmern muß.

Die Lösung für SLX bietet eine durchaus akzeptable Flexibilität, verbunden mit einem größeren Komfort bei der Modellerstellung als C++. Das SLX-HLA-Interface läßt z.B. flexible Synchronisationsmechanismen zu wie dynamische Lookahead-Werte; es erlaubt außerdem den Transfer des Eigentums von Objektattributen sowie die Nutzung diverser Transportmechanismen.

Die Lösung für Simplex bietet die Vergleichsweise kleinste Flexibilität des HLA-Interfaces, da sämtliche HLA-bezogenen Aktivitäten automatisch ablaufen. Attribut-Updates werden automatisch generiert, wenn sich die entsprechende Simplex-Variable ändert. Der Nutzer kann nicht beeinflussen, ob gewisse Updates gleichzeitig (d.h. in einem Paket) gesendet werden sollen. Dies wäre einerseits aus Performancegründen, aber möglicherweise auch aus semantischen Gründen z.T. durchaus sinnvoll. Weiterhin wird bzgl. der zeitlichen Synchronisation von einem Lookahead des Federates von Null ausgegangen, da dies für eine allgemeine Gültigkeit der kleinste gemeinsame Nenner ist. Aus Performancegründen wäre es wiederum sinnvoll, falls es das Modell erlaubt, größere Lookaheadwerte zu verwenden.

3 Referenzanwendung

Anhand der Kopplung der Simulatoren Simplex 3 und SLX konnte eine Referenzanwendung entwickelt werden, die das Potential der HLA-basierten Kopplung von Modellen in der zivilen Simulationswelt aufzeigt. In der Referenz-Federation arbeiten ein rein zeit-diskretes SLX-Modell und ein teilweise zeit-kontinuierliches Simplex-Modell zusammen. Beide stellen grundverschiedene Teilmodelle dar, die zusammen ein neues „Gesamtmodell“ formen.

Das SLX-Modell simuliert die logistischen Prozesse in einem Transportunternehmen, eine Aufgabe, für die SLX durch seine prozessorientierte Weltansicht 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 von Hause aus anbietet, ist Simplex für diese Modellierungsaufgabe prädestiniert. Mittels HLA ist es nun möglich, Teile des Gesamtmodells mit dem dafür am besten geeigneten Simulatoren 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 1 zeigt das daraus resultierende Setup der Federation.

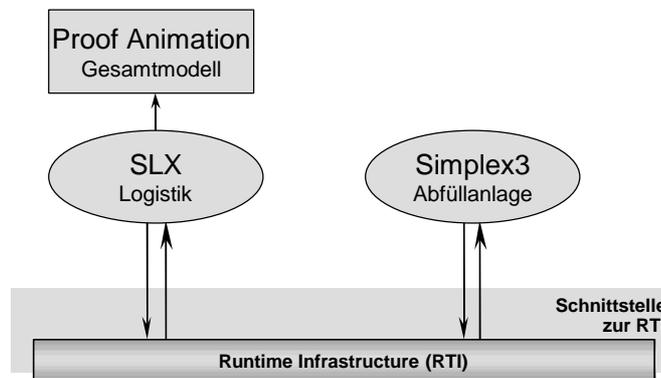


Abbildung 1: Setup der Simplex-SLX-Federation

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 (Abbildung 2).

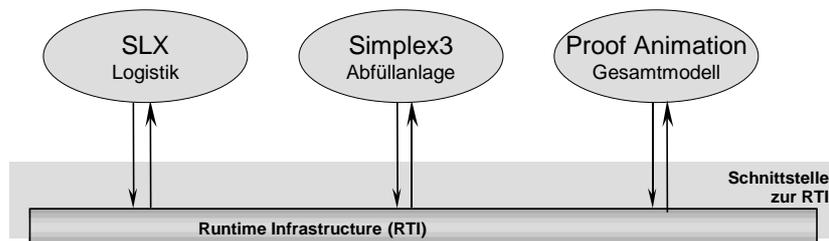


Abbildung 2: Mögliche Setup-Variante der Federation

Es ist lediglich zu beachten, daß das Federate, welches die Animation „produziert“, sämtliche für die Federation relevanten Visualisierungsinformationen empfängt. Dies bedingt möglicherweise, das anderweitig nicht notwendige Informationen „public“

gemacht werden müssen und für diese Informationen Updates zu versenden sind. Im Referenzbeispiel betraf dies z.B. die kontinuierliche Größe „Füllstand“, für die ohne die Visualisierungsoption keine Übertragung von Simplex an SLX hätte erfolgen müssen. Zur Reduzierung der zu übertragenden Daten ist es also durchaus wichtig, beim Planen der Federation zu beachten, wo und durch wen eine Visualisierung zu erfolgen hat. Einen Screenshot der für das Referenzbeispiel durchgeführten Visualisierung zeigt Abbildung 3.

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 angepaßte Kommunikationsprotokolle. Mit der Verwendung von HLA wird eine neue Stufe in der Kopplung zwischen diskreten und kontinuierlichen Modellen erreicht.

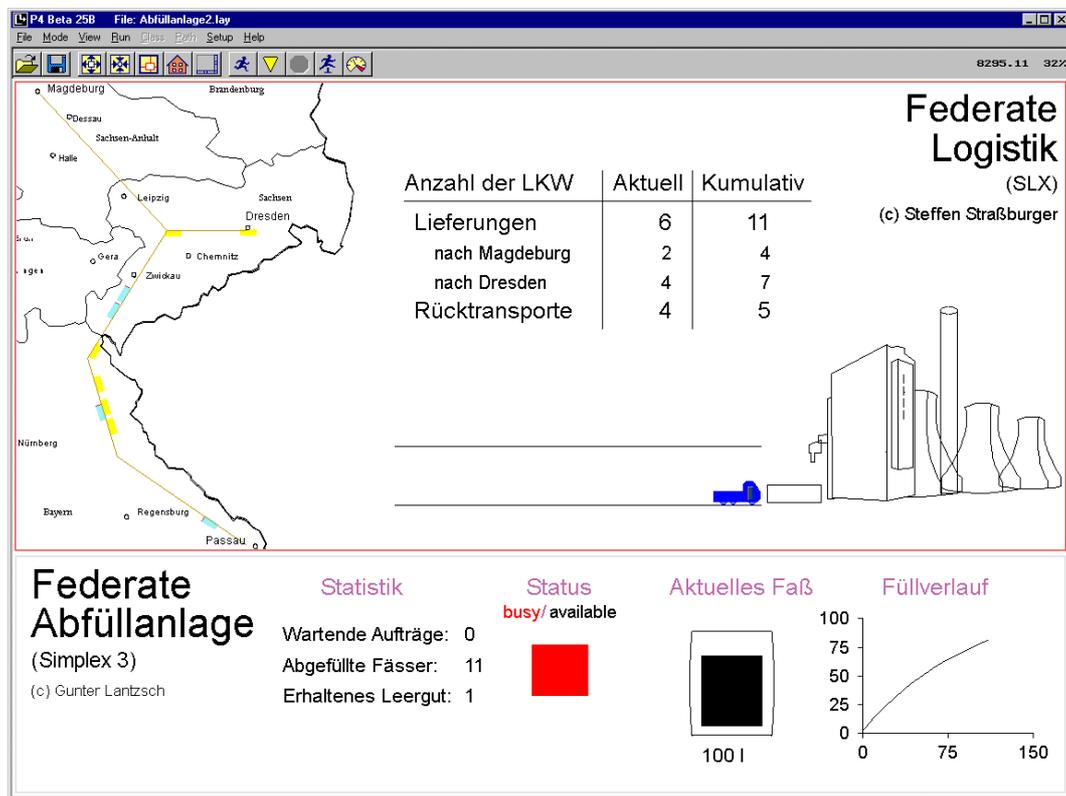


Abbildung 3: Screenshot der Online-Animation der Simplex-SLX-Federation

4 Was bringt's ?

Mit der Integration von HLA-Schnittstellen in Simulationssysteme ist es möglich, ein komplexes Simulationsmodell auf der Basis verschiedener Simulatoren zu erstellen und die Stärken des jeweiligen Simulators für eine spezielles Anwendungsgebiet zu nutzen. So zeichnet sich Simplex unter anderem durch seine progressiven Algorithmen zur

numerischen Integration zur Modellierung von kontinuierlichen Abläufen aus. SLX hingegen ist ein Simulator für diskrete Modelle, der durch seine Objektbasiertheit und seine prozessorientierte Weltsicht sehr komfortabel die Modellierung logistischer Sachverhalte ermöglicht. Eine HLA-basierte Kopplung ermöglicht nun, die Vorzüge beider Systeme miteinander zu kombinieren. An einem Beispiel wurde diese Kopplung aufgezeigt.

Es stellt sich die Frage, inwieweit HLA der anvisierte Standard für das „Plug&Play“ von Modellkomponenten bzw. Referenzmodellen (siehe auch [6]) sein kann. Dieses Ziel läßt sich nach Auffassung der Autoren nur verwirklichen, wenn Simulationssysteme den Zugang zu HLA für den Nutzer in einer automatisierten Weise bereitstellen, wie es im Simplex-Ansatz der Fall ist. Eine breite Akzeptanz und großflächige Nutzung im zivilen Bereich wird HLA nur erreichen können, wenn der Modellentwickler seine bisherige Denk- und Modellierungsweise nicht ändern muß und ihm die Arbeit mit HLA bzw. die Ansteuerung des HLA-Interfaces möglichst vollständig von Simulationssystem abgenommen wird.

Wünschenswert ist eine nutzerfreundliche Welt, in der Simulationssysteme von sich aus einen Standard zur Kopplung mit anderen Systemen unterstützen. Mit der HLA entwickelt sich ein solcher Standard. Die Hersteller von Simulatoren sind daher eigentlich aufgefordert, ihrer Bringepflicht nachzukommen. Es gibt bis dato nur wenige Simulatoren, deren Hersteller selbst ein HLA-Interface anbieten.

In einer idealen Welt kann die Implementation eines HLA-Interfaces sogar so aussehen, daß sich für den Modellierer im Prozeß der Modellerstellung keine signifikanten Änderungen ergeben. Daß diese Wunschvorstellung grundsätzlich machbar ist, konnte am Simulationssystem Simplex 3 gezeigt werden.

5 Literatur

- [1] Defense Modeling and Simulation Office (DMSO). High Level Architecture Homepage. URL <http://hla.dmsomil/>
- [2] 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.
- [3] Lantzsich, G., S. Straßburger, C. Urban. HLA-basierte Kopplung der Simulationssysteme Simplex III und SLX. Tagung Simulation und Visualisierung. 4.-5. März 1999, Magdeburg.
- [4] Straßburger, S. On the HLA-based Coupling of Simulation Tools. In *Proceedings of the 1999 European Simulation Multiconference*. Juni 1-4, 1999. Warsaw, Poland.
- [5] Klein, U., S. Straßburger. Die High Level Architecture: Anforderungen an interoperable und wiederverwendbare Simulationen am Beispiel von Verkehrs- und Infrastruktursimulationen. In *Kuhn A. und S. Wenzel (Ed.), Tagungsband 11. Symposium Simulationstechnik ASIM 97*. Dortmund, November 1997, pp. 529-534, Vieweg 1997.
- [6] 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.