

## **Erfahrungen aus der Anwendung von HLA-basierter verteilter Simulation im Nutzfahrzeugbereich**

### ***Experiences from Applying HLA-based Distributed Simulation in the Utility Vehicle Sector***

Michael Raab, Fraunhofer IFF, Magdeburg (Germany)  
Thomas Schulze, University Magdeburg, Magdeburg (Germany)  
Steffen Straßburger, Ilmenau University of Technology, Ilmenau (Germany)

**Abstract:** Distributed simulation has been known for many years as a method for the simulation of complex models. This paper presents the authors' experiences from an industrial project targeting distributed simulation in the utility vehicles sector production. Existing monolithic simulation models were extended for their reuse in a distributed environment. One project requirement was to ensure the consistency between monolithic and distributed model versions. Project results show that the frequently mentioned concerns regarding distributed simulation can be resolved successfully in industrial contexts. The gathered results of distributed simulations legitimate the necessary efforts.

## **1 Motivation**

Mit verteilter Simulation wird im Allgemeinen die gleichzeitige Ausführung von mehreren Simulationsmodellen auf einem oder mehreren Prozessoren bezeichnet (Fujimoto 2000). Als daraus resultierende Vorteile seien hier die Möglichkeiten der Wiederverwendung von existierenden Simulationsmodellen in einem erweiterten Kontext und die Kopplung von Simulationsmodellen unterschiedlicher Softwarehersteller genannt.

In der Fabrikplanung und des Weiteren im Fabrikbetrieb werden häufig Simulationsmodelle für Teilbereiche der Fabrik unabhängig von einander und oft mit unterschiedlichem Detaillierungsgrad und Anwendungsfokus entwickelt. Zur Analyse der jeweiligen Teilbereiche leisten diese einzelnen Modelle ihre gewohnten Dienste. Um die Beziehungen der Teilbereiche untereinander zu modellieren, muss ein komplexes Modell genutzt werden, denn es bestehen i. a. Feedbacks zwischen den Bereichen, d. h. ein Puffer in einem nach gelagerten Bereich kann nur belegt werden, wenn dieser über eine ausreichende freie Kapazität verfügt. Des Weiteren existieren

häufig Verbindungseinheiten, wie spezielle Lagerbereiche oder Transportvorgänge, die in den existierenden Teilmodellen nicht mit abgebildet werden.

Wie kann ein entsprechendes komplexes Modell erstellt werden? Die klassische Vorgehensweise ist die Entwicklung eines neuen monolithischen Gesamtmodells, häufig mit einem geringeren Detaillierungsgrad als die Teilmodelle. Die andere Möglichkeit ist die Kopplung der existierenden Simulationsmodelle, unter Beibehaltung des Detaillierungsgrades, einschließlich der modellierten Verbindungseinheiten zwischen den Modellen zu einem komplexen verteilten Simulationsmodell.

Dieser verteilte Ansatz hat in diesem Fall folgende Vorteile, die sich vorrangig aus der Wiederverwendung der Simulationsmodelle ergeben (Strassburger et al. 2007; Rabe und Jäkel 2001):

- Einsparung an Modellentwicklungszeit im Vergleich zur Entwicklung eines neuen monolithischen Modells,
- Beibehaltung des erreichten Detaillierungsgrades und
- Sicherung der Konsistenz zwischen den Teilmodellen und dem verteilten Gesamtmodell.

Der Beitrag beschreibt die Vorgehensweise und Erfahrungen der Autoren aus einem Projekt zur Anwendung von HLA-basierten verteilten Simulationsmodellen im Nutzfahrzeugbereich. Im folgenden Kapitel werden kurz die Grundaufgaben der verteilten Simulation erläutert und der aktuelle Stand der Nutzung der verteilten Simulation im industriellen Umfeld eingeschätzt. Anschließend erfolgt eine Beschreibung der Ausgangslage für das Projekt um daraus die notwendigen Anforderungen an das Management der verteilten Simulation abzuleiten, da dem industriellen Anwender nur eine abstrakte Sicht auf die sehr detaillierten Dienste der Basissoftware ermöglicht werden soll. Einige Implementierungsdetails werden in dem daran anschließenden Kapitel erläutert. Das verteilte Simulationsmodell erlaubt die Bestimmung von Ergebnisdaten über das komplexe System mit einem hohen Detaillierungslevel. Einige Ergebnisformen werden in dem daran anschließenden Kapitel aufgezeigt. Der Beitrag schließt mit einem Ausblick auf weitere notwendige Arbeiten.

## 2 Grundaufgaben der verteilten Simulation

Zum Erreichen der benötigten Interoperabilität zwischen den existierenden und wiederzuverwendenden Simulationsmodellen in einer verteilten Umgebung sind drei Grundaufgaben durchzuführen (Fujimoto 2000).

Zuerst ist eine Menge von den Objekten aus allen beteiligten Simulationsmodellen zu erstellen, an denen alle Modelle ein Interesse haben. Änderungen von Attributwerten dieser gemeinsamen Objekte werden an alle Interessenten publiziert. Zusätzlich ist hier die Semantik der Objekte zu vereinbaren, damit alle Teilmodelle die empfangenen bzw. gesendeten Informationen einheitlich interpretieren. Diese Notwendigkeit wird auch als semantische Interoperabilität bezeichnet.

Die zweite Grundaufgabe ist der eigentliche Datenaustausch zwischen den Teilmodellen auf der Basis von bekannten publish- und subscribe-Mechanismen.

Die Synchronisation der Teilmodelle, d.h. ein wohldefinierter Zeitfortschritt der lokalen Simulationsuhren in den Teilmodellen, umfasst die dritte Grundaufgabe. Jedes Teilmodell behält seine eigene lokale Uhr, nur muss der Zeitfortschritt mit den anderen Teilmodellen abgestimmt werden. Es existieren unterschiedliche Algorithmen, die zur Organisation verwendet werden können.

Die gegenwärtig genutzten Ansätze zur verteilten Simulation im zivilen Bereich auf der Basis von kommerziellen Simulationssystemen lassen sich in zwei Kategorien einteilen. Die erste Kategorie umfasst Lösungen, in denen Simulationsmodelle nur eines Herstellers, also homogene Simulationsmodelle, miteinander gekoppelt werden. Diese Kopplungsform setzt auf die Nutzung von simulatorspezifischen Features zur Lösung der Grundaufgaben. Dieser Ansatz ist an die proprietären Eigenschaften des verwendeten Simulators gebunden und ist nicht flexibel hinsichtlich der Integration von Simulationstools mit anderen IT-Anwendungen.

Die zweite Kategorie umfasst Lösungen, die auf einem unabhängigen Framework oder Standard zur verteilten Simulation basieren, und nicht an proprietäre Schnittstellen gebunden sind. Diese Lösungen sind hinsichtlich der Integration inhomogener Komponenten wesentlich flexibler. Bekannte Beispiele für diese zweite Kategorie sind FAMAS (Boer 2005) und HLA (IEEE 2000). Der HLA-Ansatz ist eine weltweit genutzte Architektur für die verteilte Simulation und erlaubt die Kopplung sowohl von Simulations- und Nicht-Simulationskomponenten.

Trotz der Existenz des HLA-Standards seit Ende der 90-er Jahre hat sich die Nutzung der verteilten Simulation in der Anwendungsdomäne Produktion und Logistik erst zaghafte durchgesetzt. Als Gründe werden u. a. die fehlende einfache und aufwandsarme Konfigurierbarkeit der Modellkopplung angeführt (Bernhard und Wenzel 2006). Auf der anderen Seite erfolgten in den letzten Jahren Forschungen, um die Nutzung der verteilten Simulation weiter zu verbreitern (Bernhard und Wenzel 2006; Mustafee et al. 2007) und sie z. B. im Kontext der Digitalen Fabrik zur Anwendung zu bringen (Strassburger et al. 2003). Weitere Arbeiten beschäftigen sich mit der Standardisierung von Vorgehensweisen zur Nutzung von HLA im Kontext kommerzieller Simulationstools (Taylor et al. 2007)

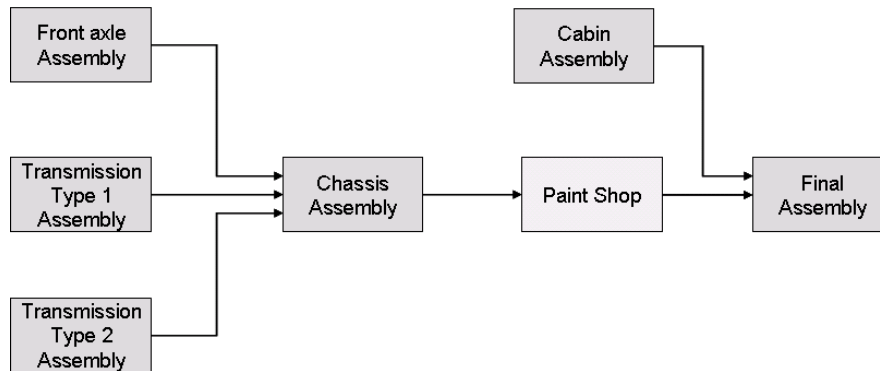
Basierend auf den erfolgten weiteren Standardisierungsarbeiten und den eigenen Erfahrungen der Autoren mit HLA wurde sich für die Nutzung dieses Standards zur Implementierung der verteilten Simulation entschlossen.

### 3 Ausgangssituation

Im Rahmen des Neubaus einer Fabrik zur Produktion von Nutzfahrzeugen wurden für sieben zusammenhängende Montage- und Farbgebungsbereiche jeweils eigenständige Simulationsmodelle entwickelt. Diese Modelle werden nach der Beendigung der Planung auch zur Unterstützung des laufenden Fabrikbetriebes verwendet. In einem konkreten Industrieprojekt bestand nun die Aufgabe, die für die einzelnen Teilbereiche der Fabrik entwickelten Modelle in ein komplexes verteiltes Modell zu integrieren. Der Materialfluss zwischen den Teilmodellen ist aus Abbildung 1 ersichtlich. Mit diesem komplexen verteilten Modell sollten die Abhängigkeiten zwischen den Teilbereichen untersucht werden. Hierzu wurden Fragestellungen der Dimensionierung der Puffer zwischen den Bereichen, die Koordinierung der Pro-

duktionspläne und die Aufdeckung von Engpässen zwischen den Bereichen hinsichtlich des Erreichens von vorgegebenen Produktionszahlen untersucht.

Das Koppeln von unterschiedlichen und unabhängig voneinander entwickelten Simulationsmodellen ist aus unserer Sicht ein typischer Anwendungsfall aus dem Bereich Produktion und Logistik. In der Vielzahl der Anwendungsfälle bezieht sich die Unterschiedlichkeit der Modelle u. a. auf die verwendete Simulationssoftware, unterschiedliche Detaillierungsgrade und Zeitfortschrittsmechanismen.



**Abbildung 1:** Teilmodelle und der Materialfluss

Die verwendeten Modelle in dem konkreten Projekt lassen sich in zwei Anwendungskategorien einteilen: Montage- und Farbgebungsmodelle. Zu jeder Kategorie existiert ein generisches SLX-Simulationsmodell, welches allgemeine Objektklassen, einschließlich der entsprechend benötigten Funktionalitäten, enthält, die zur Erstellung von konkreten, domainspezifischen Simulationsmodellen verwendet werden. Aus dem generischen Modell wird dann zur Laufzeit der Simulation ein konkretes ausführbares Simulationsmodell unter Berücksichtigung einer Konfigurationsdatei generiert. SLX ist ein erweiterbares objektbasiertes Simulationssystem (SLX 2008). Aufgrund des Vorhandenseins einer generischen HLA-Schnittstelle für den Simulator SLX in Form einer Wrapper-Bibliothek wurde HLA mit der RTI 1.3NG (Straßburger 2001) als Middleware zur Kopplung ausgewählt.

## 4 Projektanforderungen

Bei der Umsetzung des Projektes wurde von den folgenden Anforderungen ausgegangen:

- Einheitliche Modellversion sowohl für die verteilte als auch monolithische Nutzung der bestehenden Modelle,
- Einfaches Managen der mit der verteilten Simulation verbundenen Aufgaben,
- Zusätzliche Ergebnisdarstellungen, die sich nur aus dem Kontext der verteilten Simulation ergeben und
- Einhaltung von Laufzeitgrenzen

Von entscheidender Bedeutung in diesem Projekt war die Forderung, dass nur ein einziger Simulations Quellcode sowohl für die verteilte als auch für die monolithische Nutzung verwendet wird. Sowohl für die Montage- und Farbgebungsbereiche existierten zwei unterschiedliche generische Simulationsmodelle, aus denen dann mittels Konfigurationsdateien die jeweiligen spezifischen Simulationsmodelle zur Laufzeit generiert werden. Aus jedem generischen Modelltyp müssen sich sowohl verteilte als auch monolithische Modelle ableiten lassen. Nur auf dieser Grundlage ist es möglich, eine Konsistenz der Modellvarianten (monolithisch bzw. verteilt) zu gewährleisten. Die verteilten Modellvarianten müssen zusätzliche Funktionalitäten, wie beispielsweise das Reagieren auf externe Ereignisse, die Synchronisation mit den anderen verteilten Modellen und die Modellierung des Teiletransportes von einem Modell in das technologisch nachfolgende Modell enthalten. Diese zusätzlichen Funktionalitäten sind entsprechend zu kapseln.

Die an der verteilten Simulation beteiligten Modelle müssen wohldefiniert gestartet und beendet werden, um ein sinnvolles Interagieren zu ermöglichen. Auf einem sehr niedrigen Level werden von der Laufzeitumgebung der HLA die entsprechenden Dienste bereitgestellt. Für eine Nutzung der verteilten Simulation im industriellen Umfeld ist es notwendig, ein entsprechendes Nutzer-Interface zu verwenden, welches die notwendigen Dienstaufrufe koordiniert und vor dem Nutzer kapselt.

Ein Ziel der Erstellung des komplexen Simulationsmodells war die Ableitung von neuen Erkenntnissen aus dem Zusammenwirken der einzelnen Teilmodelle in einer komplexen Umgebung. Es ist deshalb notwendig, Möglichkeiten zu entwickeln, um die neuen, nur aus dem komplexen Zusammenspiel ableitbaren Ergebnisse zu erfassen und dem Nutzer zu präsentieren.

Die Laufzeit von verteilten Simulationsmodellen wird stark beeinflusst von der Größe des Lookahead und den entsprechenden Synchronisationsalgorithmen. In dem Projekt werden die Simulationsmodelle zur Unterstützung der Produktionsorganisation verwendet, wobei die Simulationsdauer i. a. 100 Tage beträgt. Die monolithischen Modelle erledigen diese Aufgabe im Sekundenbereich. Es war in dem Projekt zu sichern, das bei der verteilten Simulation, auch die Laufzeiten des verteilten Modells in akzeptablen Bereichen bleiben.

## 5 Umsetzung

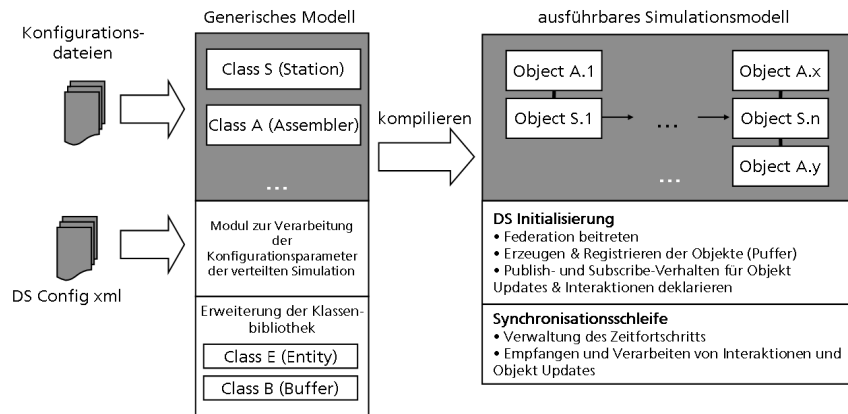
Im folgenden Kapitel werden einige Implementierungsdetails zur Erfüllung ausgewählter Anforderungen aufgezeigt.

### 5.1 Anpassung der existierenden generischen Modelle

Eine Anforderung war die Anpassung der existierenden generischen monolithischen Modelle zur Teilnahme an einer verteilten Simulationen. Das ursprüngliche generische monolithische Modell muss erweitert werden um:

- HLA-Dienste zum Beitreten zur und Verlassen einer Federation sowie dem Versenden und Empfangen von Objektattributänderungen und Interaktionen,

- die korrekte Integration von externen Ereignissen in die eigene Ereignislistenverwaltung und das Reagieren auf diese externen Ereignisse und
- die Modellierung des Teiletransportes zwischen den beteiligten Teilmodellen.



**Abbildung 2:** Anpassung der existierenden generischen Modelle für die verteilte Simulation

Die Grundidee zur automatischen Anpassung der Modelle besteht darin, die generischen Modelle mit der Fähigkeit zu versehen, bei der Generierung der spezifischen Modelle die Erweiterungen bzgl. der verteilten Ausführung mit in den generierten Code zu integrieren (Abbildung 2). Somit ist gewährleistet, dass sowohl die monolithischen als auch die verteilten Simulationsmodelle konsistent sind. Die Sprache SLX gestattet die Erweiterung des zu kompilierenden Quelltextes während der Kompilierung selbst. Durch die Nutzung dieses Features können die unterschiedlichen Varianten sehr elegant erstellt werden.

Die existierenden monolithischen Modelle zerstören die Teileobjekte, wenn diese das Ende ihrer technologischen Prozesskette erreicht haben. Wird dieses Modell in ein verteiltes komplexes Modell integriert, so muss das entsprechende Teileobjekt an das Folgemodell übergeben werden. Die Übergabebedingungen in dem beschriebenen Projekt erfordern die Übergabe der Teile in einen Puffer mit beschränkter Kapazität. Damit entspricht diese Forderung den Interoperabilitätsanforderungen des Typs A.2 aus dem Interoperability Reference Model (Taylor 2007). Die Übertragung des Teils wurde als Interaktion und ausgewählte Attribute des Puffers aus dem Folgemodell als Objekt modelliert. Abbildung 3 zeigt die Modellierung des Material- und Informationsflusses zwischen den Teilmodellen.

## 5.2 Einfluss des Lookahead

Der Lookahead gibt in der verteilten Simulation einen Zeitraum an, in dem ein Teilmodell keine Ereignisse für ein anderes Teilmodell erzeugen darf. Mit Hilfe dieser Zusicherung können die konservativen Synchronisationsverfahren eine untere Zeit-

schranke berechnen, bis zu der die Teilmodelle in ihrer lokalen Simulationszeit fortschreiten dürfen. Es gilt allgemein, dass mit steigenden Lookahead die Parallelität der Abarbeitung steigen kann und somit die Rechenzeit sinkt. Experimente mit unterschiedlichem Lookahead bestätigten diese generelle Aussage (Abb. 4). Aus Sicht der Performance ist ein Lookahead größer 0 zu wählen.

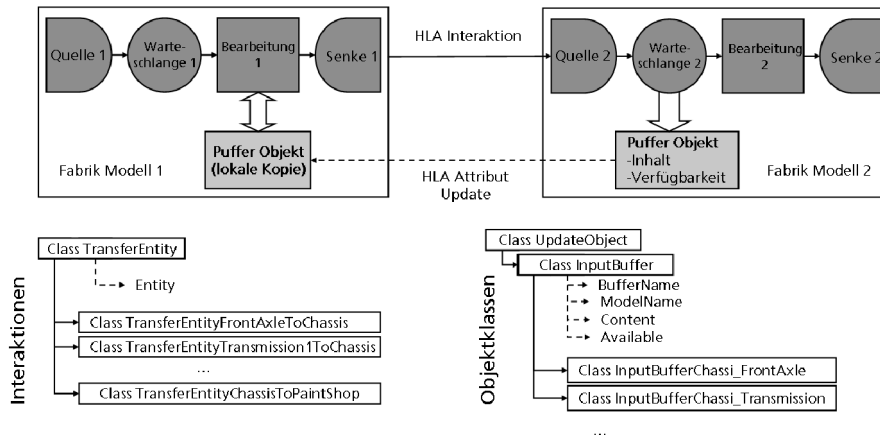
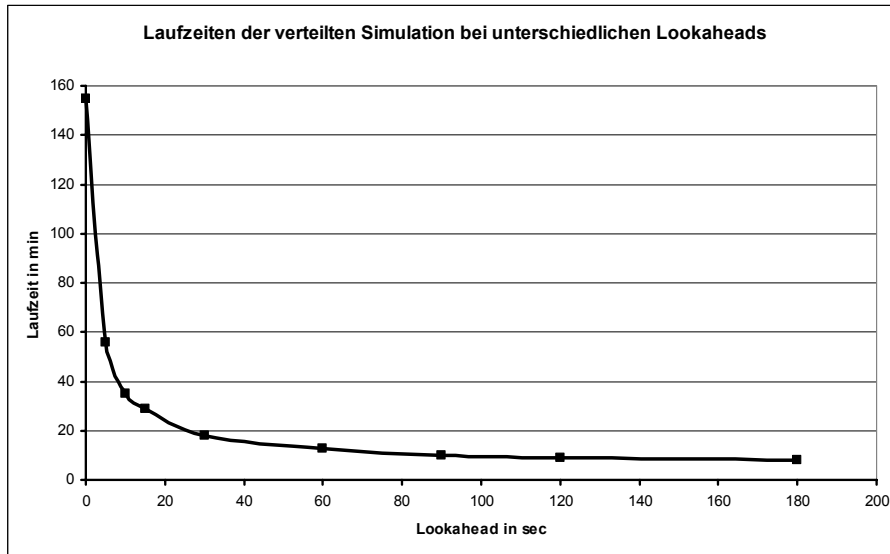


Abbildung 3: Modellierung des Material- und Informationsflusses

Ein Lookahead größer 0 bleibt nicht ohne Auswirkungen auf die Logik der Abarbeitung der Ereignisse. Dies soll an zwei Beispielen demonstriert werden. Dabei versorgt das Modell A das Modell B mit den entsprechenden Teilen. Die im Modell B ankommenden Teile werden in einem Puffer mit begrenzter Kapazität eingelagert. Wenn die Kapazitätsgrenze im Puffer erreicht wird, so ist keine weitere Aufnahme mehr möglich und Modell A kann keine Teile abgeben. Ebenso kann das Modell B die Fertigung eines Teiles erst beginnen, wenn es aus dem Modell A im Modell B eingetroffen ist.

Der Pufferinhalt wurde zum Zeitpunkt  $t_1$  im Modell B aktualisiert. Die Information über den neuen aktuellen Inhalt erreicht das Modell A zum Zeitpunkt  $t_2 = t_1 + \text{Lookahead}$ . Bis zum Zeitpunkt  $t_2$  verwendet die interne Logik im Modell A einen nicht korrekten Wert für den aktuellen Pufferinhalt. Dieser Umstand kann zu Fehlentscheidungen in der Logik führen, wenn der maximale Pufferinhalt schon erreicht worden ist, aber diese Information dem Modell A bis zum Zeitpunkt  $t_2$  nicht bekannt ist.

Ein weiteres Problem ergibt sich, wenn Modell A zum Zeitpunkt  $t_3$  ein Teil an das Modell B sendet. Die mit dem Teil verbundene Interaktion erreicht das Modell erst zum Zeitpunkt  $t_4 = t_3 + \text{Lookahead}$ . Auswirkungen auf die Logik im Modell B ergeben sich, wenn der Pufferinhalt zum Zeitpunkt  $t_3$  leer ist und das Modell B ein neues Teil aus dem Puffer in die Fertigung übernehmen will. Obwohl sich das Teil im Puffer „befindet“, bekommt das Modell B das entsprechende Ereignis erst zum Zeitpunkt  $t_4$  eingeplant.



**Abbildung 4:** Laufzeiten der verteilten Simulation in Abhängigkeit vom Lookahead

Im Rahmen dieses Projektes wurden keine verallgemeinerungsfähigen Lösungen erarbeitet. Spezielle, applikationsabhängige Ansätze wurden zur Lösung der obigen Problemstellungen verwendet.

### 5.3 Nutzerinterface

Für den industriellen Einsatz der verteilten Simulation ist es zwingend notwendig den Nutzer bei der operationalen Arbeit mit der verteilten Simulation zu unterstützen. Die existierenden Features und Nutzerinterfaces von kommerziellen Simulatoren unterstützen nicht die Modellierung und das Management von verteilten Simulationsmodellen. Vor diesem Hintergrund wurde in dem Projekt ein entsprechendes Nutzerinterface, ein Commander-Tool, entworfen und implementiert. Dieses Nutzerinterface unterstützt den Nutzer bei der Auswahl von Teilmodellen, der Konfiguration der entsprechenden Teilmodelle zu einem komplexen verteilten Modell unter Einbeziehung des entsprechenden Materialflusses zwischen den Teilmodellen, bei der Durchführung der Simulation und der Aufbereitung von Ergebnissen.

Die gegenwärtige Entwicklungsversion offeriert die folgenden Dienste:

- Kapselung der notwendigen HLA-RTI-Services gegenüber dem Nutzer,
- Unterstützung bei der Zusammenstellung des komplexen verteilten Modells,
- Starten, Steuern und Beenden der gesamten Ausführung der verteilten Simulation und
- Berechnung von Ergebnisstatistiken und -visualisierungen über die Zusammenhänge im komplexen Modell.



Der Commander wurde in C++ entwickelt um einen einfachen Zugriff auf die RTI-Services über das HLA-API zu ermöglichen. Zur Konfiguration wird eine XML-Datei verwendet.

In seiner aktuellen Aufbaustufe unterstützt der Commander den Nutzer auf einem hohen Abstraktionslevel hauptsächlich bei der Generierung der entsprechenden Federation und das Management dieser Federation zur Laufzeit.

## 6 Resultatdarstellungen

Neben der standardmäßigen numerischen Resultataufbereitung durch existierende monolithische Teilmodelle, sammelt ein dem Nutzer-Interface angekoppeltes Observer-Federate zusätzliche Daten hinsichtlich der verteilten Simulation. Dazu gehören beispielsweise die Anzahl der übertragenden Teile von einem Modell zum anderen und eine Füllstandshistorie der Pufferstände. Diese Daten werden aus dem normalen Kommunikationsverkehr der Teilmodelle gesammelt. Zusätzlich werden klassische Geschäftsgraphiken über den zeitlichen Verlauf ausgewählter Resultatgrößen automatisch generiert

Zur Charakterisierung der Beziehungen zwischen den einzelnen Fertigungsbereichen werden in den abgebenden Teilmodellen die Wartezeiten durch erreichte Pufferkapazitäten und in den empfangenden Teilmodellen die Wartezeiten durch das Fehlen von Teilen erfasst. Diese Informationen erleichtern die Engpass-Analyse.

Zusätzlich erfolgt eine Visualisierung des Zustands des komplexen Modells in einer schematischen Darstellung. In dieser post-run Animation werden die Pufferstände zwischen den Modellen und die aufsummierten Teileein- und -ausgänge angezeigt.

## 7 Weitere Arbeiten

Das Ziel des Projektes, verteilte Simulationen unter Einbeziehung von existierenden Teilmodellen im industriellen Umfeld durchzuführen, wurde erreicht. Mit dem Projekt wurde gezeigt, dass verteilte Simulation im industriellen Umfeld möglich und auch handhabbar ist.

Weiterführende Arbeiten werden in zwei Richtungen geführt. Das ist zum einen die Suche nach besseren, vor allem allgemein gültigeren Lösung des Lookahead-Konfliktes zwischen der Rechenzeit und der Genauigkeit der Ergebnisse. Die andere Richtung ergibt sich aus der weiteren Unterstützung des Nutzers beim Management der mit der verteilten Simulation verbundenen Aufgaben.

## Literatur

Bernhard, J., Wenzel, S. (2006) Verteilte Simulationsmodelle für produktionslogistische Anwendungen – Anleitung zur effizienten Umsetzung. In: Schulze, T., Horton, G., Preim, B., Schlechtweg, S. (Hrsg.): Proceedings der Tagung Simulation und Visualisierung, Magdeburg. SCS Publishing House, San Diego, S. 169-178

- Boer, C. (2005) Distributed simulation in industry. Doctoral Thesis Erasmus University Rotterdam
- DMSO. (2005) The High Level Architecture. Defence Modeling and Simulation Office, Washington DC. <http://www.hla.dsmo.mil>
- Fujimoto, R. (2000) Parallel and distributed simulation systems. Wiley-Interscience, New York
- IEEE (2000) IEEE 1516-2000 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules
- Mustafee, N., Taylor, S.J.E., Brailsford, S.; Katsaliaki, K. (2007) Using CSPI distributed simulation standards for the analysis of a health supply chain. In: Schulze, T., Preim, B., Schumann, H. (Hrsg.): Proceedings der Tagung Simulation und Visualisierung, Magdeburg. SCS Publishing House, San Diego, S. 155-168
- Rabe, M., Jäkel, F.-W. (2001) Non military use of HLA within distributed manufacturing scenarios. In: Schulze, T., Schlechtweg, S., Hinz, V. (Hrsg.): Proceedings der Tagung Simulation und Visualisierung, Magdeburg. SCS-European Publishing House, Erlangen, S. 141-150
- SLX (2008) Simulation Language with eXtensibility. Wolverine Software, Alexandria, VA, <http://www.wolverinesoftware.com>
- Strassburger, S. (2001) Distributed simulation based on the High Level Architecture in civilian application domains. Doctoral Thesis, University of Magdeburg, Germany
- Strassburger, S., Schmidgall, G., Haasis, S. (2003) Distributed manufacturing simulation as an enabling technology for the digital factory. Journal of Advanced Manufacturing Systems (JAMS) 2 (2003) 1, S. 111-126
- Strassburger, S., Schulze, T., Lemessi, M. (2007). Applying CSPI reference models for factory planning. In: Henderson, S.G.; Biller, B.; Hsieh, M.-H.; Shortle, J.; Tew, J.D.; Barton, R.R. (eds.) Proceedings of the 2007 Winter Simulation Conference, Washington, D.C. (USA). IEEE, Picataway, S. 603-609
- Taylor, S.J.E., Mustafee, N., Strassburger, S., Turner, S.J., Low, M.Y.H., Ladbrook, J. (2007). The SISO CSPI PDG standard for commercial off-the-shelf simulation package interoperability reference models. In: Henderson, S.G.; Biller, B.; Hsieh, M.-H.; Shortle, J.; Tew, J.D.; Barton, R.R. (eds.) Proceedings of the 2007 Winter Simulation Conference Washington, D.C. (USA). IEEE, Picataway, S. 594-602