

Fachwissenschaftliche Projektarbeit

Entwicklungspfade und Zukunftschancen der verteilten Simulation in der Produktion und Logistik

Autor Arne Dieckmann
Matrikelnummer 157358

Ausgegeben am 19.08.2015
Eingereicht am 8.12.2015

Betreuer Univ.-Prof. Dr.-Ing. Markus Rabe
Dipl.-Geoinf. Maik Deininger

Inhaltsverzeichnis

1. Einleitung.....	1
2. Grundlagen der Simulationstechnik.....	2
2.1 Simulationsarten.....	4
2.1.1 Kontinuierliche Simulation.....	4
2.1.2 “Time Slicing Approach”	5
2.1.3 Ereignisdiskrete Simulation.....	5
2.2 Verteilte und parallele Simulation.....	6
2.3 Technischer Hintergrund	8
2.3.1 High Level Architecture.....	9
2.3.2 CORBA.....	11
2.3.3 GRIDS.....	11
2.3.4 GRID Computing.....	11
2.3.5 Multiagentensysteme.....	12
2.4 Konzeptionelle Herangehensweise bei der Realisierung einer Simulation.....	12
2.4.1 Konzept- Modellierung.....	12
2.4.2 Klassifizierung der Daten.....	14
2.4.3 Simulationsansätze.....	15
2.4.4 Erstellen eines Computermodells.....	16
2.5 Herausforderungen der verteilten Simulation.....	16
2.5.1 Synchronisation von Prozessen.....	16
2.5.2 Zeitkonsistenz.....	17
3. Anwendungspotenzial verteilter Simulation.....	19
3.1 Erste Forschung für zivile Anwendungen.....	19
3.2 Analysen der potentiellen Anwendungsgebiete in der Logistik.....	21
3.2.1 Vorteile verteilter Simulation gegenüber normaler Simulationssoftware.....	21

3.2.2 Herausforderungen bei praktischer Anwendung von verteilter Simulation.....	22
3.3 Anwendungen verteilter Simulation.....	23
3.3.1 Anwendungsbeispiele Logistik.....	23
3.3.2 Anwendungsbeispiele in der Produktion.....	26
3.3.3 Verkehrssimulation.....	28
3.3.4 Virtuelle Trainingsumgebungen.....	29
3.4 Zukunftspotenziale verteilter Simulation	30
4. Entwicklungspfade und Kategorisierung der Forschung zur verteilten Simulation.....	32
4.1 Herausforderungen bei der Datenanalyse.....	33
4.2 Gliederungen der Forschungsgebiete	34
4.2.1 Logistische Anwendungen.....	34
4.2.2 Verteilte Simulation in der Produktion.....	35
4.2.3 Verkehrssteuerung.....	36
4.2.4 Fabrik- und Serviceplanung.....	37
4.2.5 Militärische Anwendungen.....	37
4.2.6 Virtuelle Umgebungen.....	37
4.2.7 Interdisziplinäre Beiträge.....	38
4.2.8 Interoperabilität.....	38
4.2.9 Performanceanalyse.....	39
4.2.10 Forschung zur grundlegenden IT-Architektur.....	39
4.3 Auswertung und Schlussfolgerungen.....	40
Literaturverzeichnis.....	42
Abbildungsverzeichnis.....	45
Abkürzungsverzeichnis.....	46
Anhang (Referenztable für Kapitel 4)	

Kapitel 1: Einleitung

Simulation ist schon über Jahre hinweg ein wichtiges Themen, das in der Wirtschaft und Wissenschaft seine Anwendung findet, da aufgrund von immer komplexer werdenden Systemen, wie einer Produktionsanlage oder einem Verkehrsnetz, Möglichkeiten gefunden werden müssen diese zu testen, ohne sie real auszuführen. Diese fachwissenschaftliche Projektarbeit beschäftigt sich mit einem Teilgebiet der Simulationstechnik, der sogenannten verteilten Simulation. Diese spezielle und neuere Form der Simulationstechnik bietet Möglichkeiten komplexe Systeme zu simulieren und ein Ziel dieser Projektarbeit ist es die Chancen und das Potenzial der verteilten Simulation zu analysieren.

Ausgehend von den Grundlagen der Simulationstechnik (Kapitel 2) sollen Anwendungsbeispiele präsentiert werden, in welchen verteilte Simulation eingesetzt wurde und einen adäquaten Modellierungsansatz darstellt. Auch werden die technischen Herausforderungen, die Vorteile von verteilter Simulation gegenüber „normaler“ Simulationssoftware und das Zukunftspotenzial thematisiert.

Das letzte Kapitel dieser Projektarbeit präsentiert dann eine Analyse der Publikationen zum Thema verteilter Simulation, sodass Entwicklungspfade bestimmt und potenzielle Entwicklungen eingeschätzt werden können. Auf Grundlage von wissenschaftlichen Beiträgen der Fachtagungen der ASIM sowie der Winter Simulation Conference (WSC) wird eine Kategorisierung erstellt und die Hauptthemen werden präsentiert. Die vorgenommene Literaturanalyse soll die Forschungsschwerpunkte darstellen, um so die Zukunftschancen verteilter Simulation, insbesondere in der Produktion und der Logistik, zu untersuchen.

Kapitel 2: Grundlagen der Simulationstechnik

Der Begriff Simulationstechnik ist sehr weit gefasst, denn es gibt ein sehr großes Spektrum an Aspekten, die bei dem Erstellen von Simulationen berücksichtigt werden müssen. Ausgehend von den Grundlagen der Simulationstechnik wird verteilte Simulation erläutert, sodass die Unterschiede zu herkömmlichen Modellierungsarten erkennbar und der Nutzen ersichtlich wird.

Vor der Umsetzung eines neuen Projektes, beispielsweise des Neubaus einer Fabrik, Planungen von Routen in Logistikunternehmen oder der Austestung von Kapazitätsgrenzen bestimmter Dienstleistungsunternehmen (Banken, Flughäfen, Restaurants etc.) stellt sich die Frage, wie die ablaufenden Prozesse optimiert werden können. Es gibt drei Möglichkeiten, diese kritischen Fragen zu klären, denn man kann zum einen ein Projekt ohne vorangegangene Planungen umsetzen und überprüft erst hinterher die Leistungsfähigkeit des Systems. Zum anderen können Erfahrungs- und Vergleichswerte bekannter Projekte herangezogen und diese als Grundlage für die Umsetzung genutzt werden. Drittens besteht die Möglichkeit, das reale System zu simulieren (Robinson, 2004). Den Begriff Simulation grenzt Robinson (2004) folgendermaßen ein:

„Experimentation with a simplified imitation (on a computer) of an operations system as it progresses through time, for the purpose of better understanding and/ or improving that system.“

Seine Definition beinhaltet alle wichtigen Schlagworte und zunächst sollten diese genauer geklärt werden. Imitation bedeutet „mimicking or copying something else“ (Robinson, 2004), was heißt, dass mit Hilfe von Vereinfachungen ein reales System in eine computergesteuerte Simulation überführt wird. Nach Wild (2002) ist ein „Operating System“ etwas, an dem Dienstleistungen oder Produkte umgesetzt werden, also die Umgebung, die simuliert werden soll. All das wird vor dem Hintergrund durchgeführt, das System besser zu verstehen und Verbesserungen zu implementieren (Robinson, 2004). Da vor einer Simulation noch nicht bekannt ist, wie ein System genau reagiert und welchen Einfluss die einzelnen Inputfaktoren haben, beinhalten computergesteuerte Simulationen Experimente, die durch Variation sämtlicher Parameter die Planung eines Systems unterstützen. Die Simulation liefert also Erkenntnisse darüber, welchen Einfluss die Inputfaktoren haben und somit ist es besser möglich Werte bestimmter Parameter für das reale System festzulegen.

Simulationen werden heutzutage in allen Bereichen eingesetzt, wobei der Fokus nicht auf statischen Systemen liegt, die die zugrunde liegende Umgebung nur zu einem konkreten Zeitpunkt betrachtet, sondern auf Systemen, die sich kontinuierlich verändern. Diese „dynamischen Simulationen“ (Law und Kelton, 2000) umfassen sowohl physische Umgebungen wie Produktionsanlagen, als auch Systeme, die maßgeblich durch Menschen beeinflusst werden (Robinson, 2004).

Da das Kernanliegen von computergesteuerter Simulation das Lösen von komplexen Problemen ist, treten viele Faktoren auf, die es schwierig machen, eine Simulation zu modellieren. Nach Robinson (2004) gibt es einige Herausforderungen:

1) Variabilität

Da die simulierten Systeme nicht statisch sind, muss eine Simulation immer so flexibel sein, dass sie auf Veränderungen reagieren und deren Auswirkungen richtig kalkulieren kann. Die Simulation sollte vorhersehbare Variabilität hinreichend modellieren können und ebenso müssen unvorhersehbare Änderungen darstellbar sein können. Ein Beispiel für vorhersehbare Variabilität wäre die Änderung der Angestelltenanzahl in einem Betrieb.

2) Prozesse sind gekoppelt

In der Regel sind alle Komponenten einer Simulation miteinander gekoppelt und insbesondere Variabilität macht dies zu einem Problem. Besitzt man beispielsweise drei Maschinen, die hintereinander einen Auftrag bearbeiten, treten keine Probleme auf, solange das Eingehen der Aufträge keiner Variabilität unterliegt. Sobald Aufträge zu einem nicht konkret vorhersehbaren Zeitpunkt eintreffen, wird die Planung weitaus schwieriger, weil es zu Leerständen oder Überbelastungen kommen kann. Die Aufträge beeinflussen sich gegenseitig und somit ist es kaum möglich vorherzusagen, wie schnell ein Auftrag bearbeitet werden kann.

3) Komplexität

Simulation beschäftigt sich mit komplexen Problemen und am besten kann man dies mit einem Beispiel illustrieren. Wir gehen von einem „Job-Shop System“ aus, welches ein Modell für Maschinenbelegungen ist (Gabler Wirtschaftslexikon). Es wird festgelegt, dass auf m Maschinen a Aufträge ausgeführt und dazu genau s Arbeitsschritte benötigt werden. Der Einfachheit halber gehen wir nun davon aus, dass in einem Auftrag das Produkt genau einmal von jeder Maschine bearbeitet werden muss, wobei die Reihenfolge irrelevant ist. Die Arbeitsschritte s entsprechen also immer der Anzahl der Maschinen m .

Bei zwei Maschinen gestaltet sich das Problem für den ersten Arbeitsauftrag noch übersichtlich:

- 1) $M1 \rightarrow M2$ oder
- 2) $M2 \rightarrow M1$

Bei drei Maschinen sieht es wie folgt aus:

- 1) $M1 \rightarrow M2 \rightarrow M3$ oder
- 2) $M1 \rightarrow M3 \rightarrow M2$ oder
- 3) $M2 \rightarrow M1 \rightarrow M3$ oder
- 4) $M2 \rightarrow M3 \rightarrow M1$ oder
- 5) $M3 \rightarrow M1 \rightarrow M2$ oder
- 6) $M3 \rightarrow M2 \rightarrow M1$

Offensichtlich kalkuliere sich die Anzahl der Möglichkeiten nach der Formel $m!$, sodass schon an so einem trivialen Beispiel zu erkennen ist, wie schnell ein Problem komplex wird. Somit zeigt sich, dass Systeme oft so umfangreich sind, dass nicht alle Optionen nur durch Nachdenken erfasst werden können und rechnergesteuerte Simulationen besser geeignet sind, um adäquate Vorhersagen treffen zu können. Des Weiteren neigen wir als Menschen schnell dazu, Entwicklungen, die auf den ersten Blick unwichtig erscheinen, zu unterschätzen.

Gehen wir davon aus, dass ein Modell für einen Flughafen erstellt werden soll, der in einem Gebiet gebaut wird, in dem von einem Passagierzahlanstieg von 0,1% pro Monat zu rechnen ist. Diese Veränderung scheint vernachlässigbar klein zu sein, aber beim Nachrechnen für einen Zeitraum von fünf Jahren würde dies zu einem Wachstum von 6,18% führen ($1,001^{60}=1,0618$), was für einen Flughafen sicherlich ein signifikantes Wachstum wäre. Anhand dieser zwei äußerst einfachen Beispiele sieht man, wie schnell auch geringe Änderungen zu extremen Auswirkungen hinsichtlich der Komplexität führen, sodass jedem Faktor ausreichend Beachtung geschenkt werden sollte.

Weitere Probleme der computergesteuerten Simulation sind nach Robinson (2004):

- 1) Hohe Kosten für die Simulationssoftware
- 2) Zeitaufwand für eine adäquate Simulation
- 3) Fehlerhaften Resultate der Simulation
- 4) Sehr große Datenmenge erforderlich (gegebenenfalls müssen diese zunächst empirisch ermittelt werden)

Dem gegenüber stehen folgende Vorteile:

- 1) Die Kreativität wird signifikant erhöht, da alle möglichen Ideen für Effizienzsteigerungen ausprobiert werden können, ohne dass man das Risiko eingeht der Firma zu schaden. In der Realität kann nur in einem sehr begrenzten Rahmen „ausprobiert“ werden.
- 2) Da die Entwickler einer Simulation sich intensiv mit den Abläufen beschäftigen, haben sie schnell ein besseres Verständnis des Systems. Sie können so leichter Ergebnisse prognostizieren oder Änderungen von Inputfaktoren auswerten.
- 3) Man kann das System visualisieren, was dem Verständnis dient und gegebenenfalls Schwächen aufdeckt.

2.1 Simulationsarten

Nachdem das Thema Simulation allgemein vorgestellt wurde, sollen nun einige zentrale Begriffe erläutert werden, damit klar der Unterschied von verteilter Simulation zu anderen Simulationsarten zu erkennen ist.

2.1.1 Kontinuierliche Simulation

Diese Simulationsart ist geeignet für Systeme, die sich, wie der Name schon sagt, kontinuierlich über die Zeit ändern. Für solche Systeme finden sich sehr viele Beispiele, insbesondere in der Natur. Ein Beispiel wäre die Modellierung des Wasserstandes in einem Behälter. Zustandsänderungen in diesen Systemen können mit Hilfe von Differentialgleichungen beschrieben werden und kontinuierliche Simulation bietet einen Ansatz solche Systeme zu modellieren, indem man sie numerisch löst (Özgün und Barlas, 2009). Nicht alle Systeme, insbesondere Modelle für Produktionsanlagen oder Supply Chains, ändern sich wie gerade beschrieben, sondern sie werden durch Ereignisse bestimmt, die zu einem bestimmten Zeitpunkt eintreten (siehe Unterkapitel 2.1.3 „Ereignisdiskrete

Simulation“). Es ist zu erkennen, dass die Zeit bei Simulationen eine zentrale Rolle spielt und diese wird nun im folgenden Ansatz beschrieben.

2.1.2 “Time Slicing Approach”

Diese Methode teilt die simulierte Zeit in *konstante* Zeiteinheiten Δt und man modelliert zu jedem Zeitpunkt t_i die Situation im System (Robinson, 2004). Das große Problem hierbei ist, dass man mit äquidistanten Zeitabschnitten rechnet, sodass eventuell gar keine Veränderung zwischen zwei betrachteten Zeitpunkten auftritt. Dies führt zu vielen unnötigen Berechnungen und zudem muss vorher Δt definieren werden, was insbesondere bei einer großen Variabilität der Ereignisse schwierig ist. Bei einer kontinuierlichen Simulation, welche in Kapitel 2.1.1 erläutert wurde, ist dies aber ein Lösungsansatz, da sehr eindeutig die Veränderungen im System beobachtet werden können. Δt beeinflusst maßgeblich die Genauigkeit und den Rechenaufwand für eine Simulation und muss daher den Anforderungen entsprechend gewählt werden. Fujimoto (2000) bezeichnet den Ansatz mit äquidistanten Zeitabschnitten als „Time-Stepped Simulation“.

2.1.3

Ereignisdiskrete Simulation

Der „Discrete-Event Simulation Approach“ (Robinson, 2004) unterteilt die Zeit etwas anders, denn es werden keine äquidistanten Zeitabstände gebildet, sondern es werden lediglich die Zeitpunkte berücksichtigt, in denen eine Veränderung im System auftritt. Betrachtet werden konkrete, zeitlich exakt definierte Ereignissequenzen. Somit sind nur die Zeitpunkte der Ereignisse entscheidend, wobei man diesen Ansatz noch mit der „Three-Phase Simulation Method“ (Robinson, 2004) erweitert hat. Dabei werden die Ereignisse in B (bound oder booked) und C (conditional) unterteilt. Der Unterschied dabei ist, dass B-

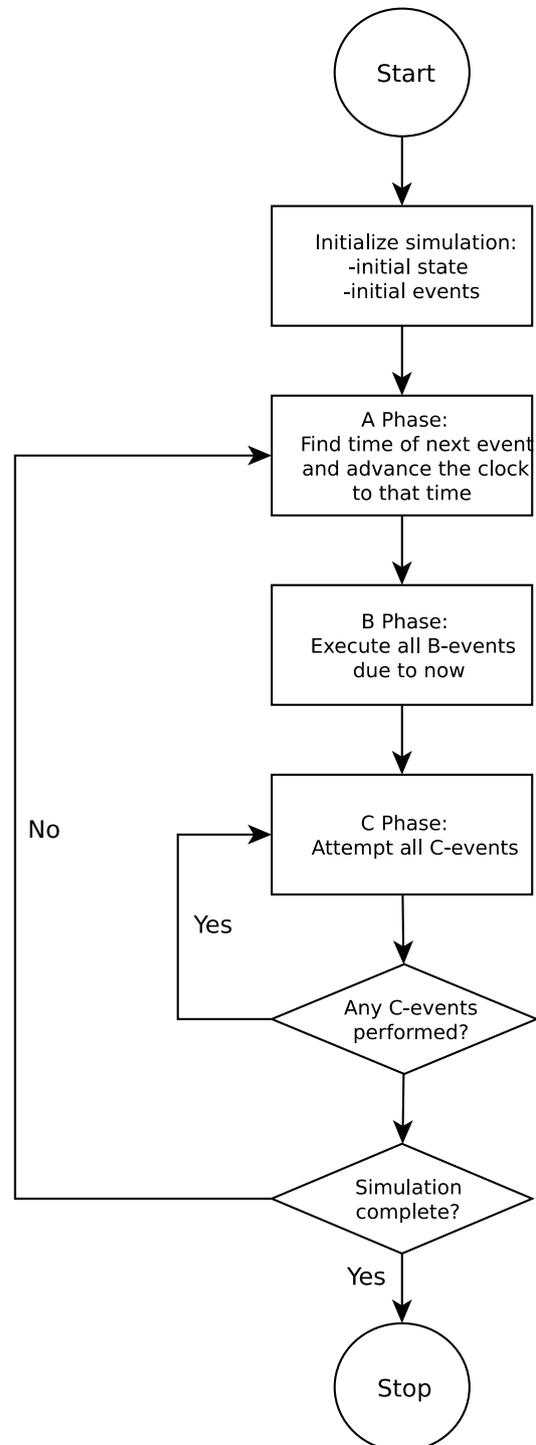


Abbildung 1: Drei-Phasen Methode, Robinson 2004

Ereignisse direkt eine Veränderung in dem betrachteten System bewirken. Das Eintreten von C-Ereignissen ist hingegen von dem Zustand des Systems abhängig, sodass solche Ereignisse nur unter bestimmten Umständen eintreten. Man erstellt zunächst eine Liste mit allen B- und C Ereignissen und in der A- Phase muss man zuerst das nächste Event bestimmen, das auftreten wird und betrachtet diesen Zeitpunkt. In der A-Phase rückt man zeitlich zum nächsten auftretenden Ereignis vor. In der B-Phase werden alle B-Ereignisse, die bis zu diesem Zeitpunkt fällig sind, ausgeführt, während die C-Phase dasselbe für C-Ereignisse tut. Die Abbildung 1 visualisiert die „Drei-Phasen Methode“.

2.2 Verteilte und parallele Simulation

Fujimoto definiert in seinem Buch „Parallel and Distributed Simulation Systems“ aus dem Jahr 2000 diesen Ansatz folgendermaßen:

„Simply stated, this is a technology that enables a simulation program to be executed on parallel/distributed computer systems, namely systems composed of multiple interconnected computers.“ (Fujimoto, 2000)

Entscheidend bei diesem Ansatz ist, dass eine Gesamtsimulation nicht nur auf einem Rechner durchgeführt, sondern auf verschiedenen Computern realisiert wird. Somit werden auf vielen verschiedenen Rechnern die einzelnen Teilsimulationen berechnet. Der Unterschied zwischen verteilter und paralleler Simulation ist, dass verteilte Simulation davon ausgeht, dass die Rechner auch *geographisch* verteilt sind, während sich bei der parallelen Simulation die Computer an einem Ort befinden (Fujimoto, 2000). Geographisch verteilt kann bedeuten, dass die Rechner sich im selben Gebäude oder in verschiedenen Ländern befinden, solange sie nicht direkt miteinander verbunden sind. Eine weitere Abgrenzung ist, dass bei der parallelen Simulation die Rechner in der Regel „homogen“ sind oder zumindest vom selben Hersteller produziert wurden. Auch dienen diese parallel arbeitenden Rechner einem vorher festgelegten Zweck, während bei der verteilten Simulation teilweise Hardware verwendet wird, die nicht speziell für schnelle Datenübertragung ausgelegt ist (Fujimoto, 2000). Diese Effekte erhöhen die Latenzzeit bei der Datenübertragung, was eines der größten Probleme der verteilten Simulation darstellt. Dort hat man neben größeren Distanzen häufig auch inhomogene Computer, was deutlich messbare Auswirkungen auf die Performance, also die Simulationsgeschwindigkeit hat (Fujimoto, 2000).

Ergänzend dazu ist festzustellen, dass viele Systeme heutzutage zu komplex sind, um sie in einer „monolithischen“ Simulation, das heißt einem Modell, das alle Komponenten erfasst, zu modellieren. Daher geht der Trend zu Modularisierung der Simulationsmodelle (Schulze et al., 1998). Modularisierung wird von Schulze et al. (1998) so beschrieben:

„Unter verteilter Simulation wird eine zeitlich parallele Nutzung von gegebenenfalls heterogenen Simulationsmodellen auf unterschiedlichen Rechnern verstanden. Komplexe Modelle werden in eigenständige Simulationsmodelle aufgeteilt, wobei jedes selbständige Modell seinen eigenen Speicherbereich, seinen eigenen Prozessor und sein eigenes Zeitmanagement (z.B. seine eigenen Ereignislisten) besitzt.“

Verteilte Simulation soll also dabei helfen, dass heterogene Simulationsmodelle erstellt werden können und diese dann miteinander interagieren können. Komplexe Systeme müssen dann nicht einem monolithischen Modell realisiert werden, sondern werden in einfachere und anpassbare Teilsimulationen aufgeteilt. Dies hat den positiven Effekt, dass bereits erstellte Modelle wiederverwendet werden können. Die geeignete Infrastruktur zur Realisierung der Interoperabilität wird im Kapitel „Technischer Hintergrund“ beschrieben.

Zu Beginn wurde diese Technologie im militärischen Bereich eingesetzt und erforscht, um möglichst rechenintensive Simulationen zu realisieren. Man wollte beispielsweise virtuelle Welten erschaffen, in denen dann Gefechte mit Simulatoren geführt werden können. Neben den signifikant hohen Rechenleistungen dieser Simulationen befinden sich die Teilnehmer oft nicht an demselben Ort, sodass verteilte Simulation durch die geographische Verteilung der Simulationsteilnehmer nötig wurde, damit viele Personen in einer Simulation eingebunden werden können, ohne dass sie physisch an einem Ort versammelt sind. Fujimoto (2000) ordnet den Beginn der Forschung auf diesem Gebiet grob in den 80er Jahren ein. Er beschreibt zudem einige Vorteile dieser Technologie, denn bei rechenintensiven Simulationen kann es vorteilhaft hinsichtlich der Performance sein, Teilsimulationen auf verschiedenen Rechnern ablaufen zu lassen. Außerdem ist es möglich, stetig weitere Personen an einer Simulation teilnehmen zu lassen, ohne dass das System an seine Grenzen stößt. Es ist sinnvoll, in virtuellen Welten Teilnehmer einzubinden, die sich an verschiedenen Orten befinden, sodass sie direkt miteinander interagieren können. Auch fallen oft Daten, die sich gegenseitig beeinflussen, an unterschiedlichen Orten an, sodass der Nutzen und die Vorteile verteilter Simulation schnell klar werden. Fujimoto erläutert dies an dem Flugraum der USA: Im Fall einer Sturmwarnung für einen bestimmten Flughafen gibt es zwei Möglichkeiten zu reagieren, denn man kann zum einen die Flugzeuge starten lassen, was gefährlich sein und durch die schlechten Bedingungen zu einem höheren Kraftstoffverbrauch führen kann. Zum anderen könnten die Flüge gestrichen werden, was zu Verspätungen führen würde. Beide Entscheidungen hätten großen Einfluss auf alle anderen Flughäfen der USA (Anschlussflüge, Starts/Ladungen) und zudem muss eine Entscheidung sehr schnell getroffen werden. Verteilte Simulation, die auf die Daten aller Flughäfen zugreift, könnte alle möglichen Szenarien simulieren.

Nach Fujimoto ist ein weiterer Vorteil, dass sich verteilte Simulationen besser integrieren lassen. Hier könnte man zur Erläuterung das Beispiel einer Flugsimulation aufgreifen, denn die Teilnehmer fliegen vielleicht unterschiedliche Flugzeugtypen (Boeing- oder Airbusmodelle). Es wäre kosteneffizienter, eine virtuelle Welt zu erschaffen, in der die einzelnen Teilsimulationen ausgeführt und miteinander interagieren können. Die Simulationsteilnehmer hätten somit eine bessere Trainingsumgebung, da sie sehr vielseitig ist und viele individuell handelnde Personen einbindet. Zudem haben verteilte und parallele Simulationen eine höhere Toleranz gegenüber einem Systemausfall, denn wenn ein Prozessor defekt ist, können andere Prozessoren den Rechenaufwand übernehmen.

Gängig bei der verteilten Simulation ist die Anwendung zur Erzeugung von virtuellen Welten; allerdings gibt es auch eine Vielzahl von möglichen zivilen Anwendungsgebieten für analytische Simulationen. Die konkreten Anwendungen und Entwicklungspfade von verteilter

Simulation, auch bei analytischen Simulationen, werden im Folgenden ausführlich dargestellt. Zunächst sollte man diese beiden Begriffe voneinander abgrenzen.

1) Analytische Simulation

Bei dieser Simulation nutzt man quantitative Daten und versucht anhand dieser eine reale Situation zu simulieren. Dabei greift der Mensch nicht direkt in die Simulation ein, denn er legt beispielsweise nur die Inputfaktoren fest oder ändert andere Einflüsse. Statistische Verteilungen werden in der Regel der Simulation zugrunde gelegt, die beispielsweise Abweichungen von einem erwarteten Mittel einiger Parameter darstellen. Ziel ist es, lange Zeiträume und insbesondere die Zukunft zu simulieren, um das reale System auf Basis dieser Erkenntnisse anzupassen (Fujimoto, 2000).

2) Virtuelle Umgebungen (VE)

Virtuelle Umgebungen sollen die Realität möglichst detailgetreu durch eine Simulation dargestellt, in die die Menschen direkt eingreifen können. In Echtzeit beeinflusst der Mensch viele Faktoren durch konkrete Steuerung. Beispiele sind Flugzeugsimulationen oder Videospiele. Diese Simulationen werden in der Regel vor dem Hintergrund von Kostenersparnis durchgeführt (Fujimoto, 2000).

2.3 Technischer Hintergrund

Verteilte Simulation ist ein komplexes Forschungsgebiet und es wurden Standards definiert, die bei der Realisierbarkeit und Anwendbarkeit dieser Technologie helfen sollen. Es werden zunächst einige grundlegenden Begriffe wie Multiagentensystem oder die High Level Architecture erläutert. Diese spielen eine zentrale Rolle in der Forschung zur verteilten Simulation, wobei das Augenmerk auf den allgemeinen Standard HLA zu legen ist. Des Weiteren werden einige technische Probleme der verteilten Simulation dargestellt, welche Herausforderungen für die Praxis darstellen. Zunächst einmal wird aber die Hardware kurz

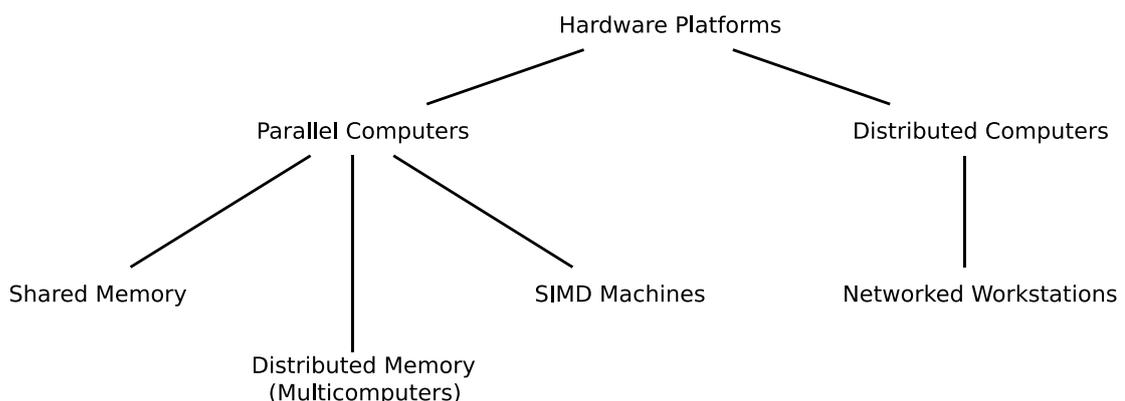


Abbildung 2: Hardware zur Realisierung verteilter und paralleler Simulation, Fujimoto 2000

vorgestellt, die für die technische Umsetzung nötig ist. Die Abbildung 2 visualisiert dies. Die parallelen Computer teilen sich nach Robinson (2004) in Shared-Memory Multiprozessoren, Distributed-Memory Multicomputer und SIMD Maschinen (single-instruction-stream,

multiple-data-stream) ein, welche in dieser Projektarbeit nicht im Detail beschrieben werden sollen, da dies zu technisch werden würde und hier mehr die Anwendungen und die Forschung in Vordergrund stehen sollen. Vielmehr soll anhand dieser Grafik gezeigt werden, dass es für parallele Simulation schon ausgereifte Hardware gibt, wobei bei der verteilten Simulation die heterogenen Computer oder leistungsfähigere „Workstations“ per LAN bei geographischer Nähe miteinander verbunden sind. Im Folgenden werden einige Begriffe erläutert, die ermöglichen, dass Teilsimulationen in einer Gesamtsimulation vereint werden können.

2.3.1 High Level Architecture

Um einheitliche Standards und ein allgemein gültiges Designprinzip für verteilte Simulation zu erzeugen, wurde die High Level Architecture (HLA) festgelegt. Deren Ziel ist, die Kompatibilität verschiedener Simulationen zu erhöhen. Man bezeichnet zunächst die Gesamtsimulation, die bekanntlich aus vielen kleinen Teilsimulationen besteht, als Föderation (engl.: federation). Die einzelnen Teilnehmer, die die Einzelsimulationen ausführen, werden Federates genannt. Die zwischengeschaltete RTI (Run-Time- Interface) verteilt die Daten zwischen den Federates und realisiert die Kommunikation (Straßburger und Klein 1997). Hierbei ermöglicht das RTI insbesondere eine „Time Stamped Ordered Message Synchronization“ (Rabe, 2000) und somit wird die Einhaltung der zeitlichen Korrektheit und Eindeutigkeit gewährleistet. Folglich stellt das RTI die zentrale zwischengeschaltete „Middleware“ dar und sorgt sowohl für Datenaustausch als auch die Koordination der Federates einer Gesamtsimulation. Die Idee, die HLA einzuführen, geht auf das US-Militär zurück und seit 1995 gibt es erste Entwicklungen dieses Standards (Fujimoto, 2000). Die HLA umfasst „Non-Runtime“ und „Runtime-Komponenten“, wobei erstere Objekte sind, die mit ihren festgelegten Attributen die reale Welt darstellen. Die „Runtime-Komponenten“ erlauben es, dass die einzelnen Federates miteinander interagieren können, damit unterschiedliche Teilsimulationen Einfluss aufeinander haben. Die Softwareumgebung, die nötig ist, um die Federates miteinander interagieren zu lassen, wird durch die RTI bereitgestellt (Fujimoto, 2000). Die Abbildung 3 visualisiert die grundlegende Technik der HLA.

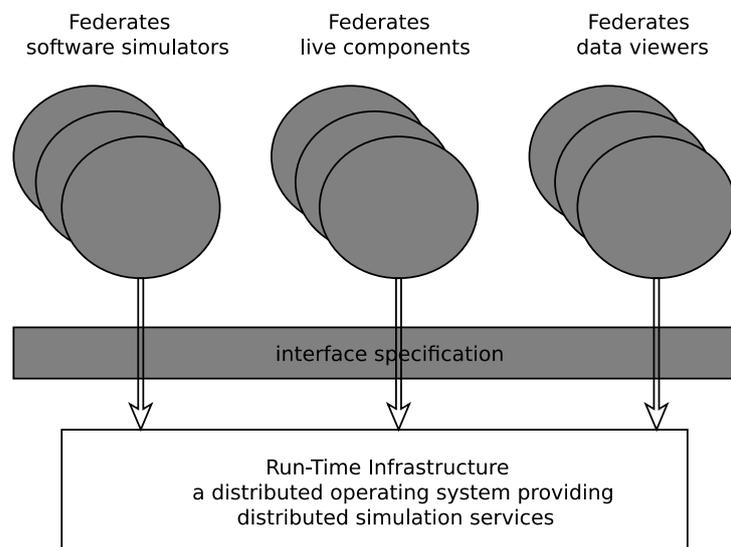


Abbildung 3: Technik der High Level Architecture, nach Fujimoto 2000

HLA RULES	
1.	Federation shall have an HLA Federation Object Model (FOM) documented in accordance with HLA Object Model Template.
2.	In a federation, all simulation-associated object instance representation shall be in the federates, not in the runtime infrastructure (RTI).
3.	During a federation execution, all exchange of FOM data among federates shall occur via the RTI.
4.	During a federation execution, federates shall interact with the RTI in accordance with the HLA Interface Specification (IFSpec).
5.	During a federation execution, an attribute of an instance of an object shall be owned by at most one federate at any time.
6.	Federates shall have an HLA Simulation Object Model (SOM), documented in accordance with the HLA Object Model Template (OMT).
7.	Federates shall be able to update and/or reflect any attributes of objects in their SOMs and send and/or receive SOM interactions externally as specified in their SOMs.
8.	Federates shall be able to transfer and/or accept ownership of attributes dynamically during a federation execution, as specified in their SOMs.
9.	Federates shall be able to vary the conditions (e.g. thresholds) under which they provide updates of attributes of objects, as specified in their SOMs.
10.	Federates shall be able to manage local time in a way that will allow them to coordinate data exchange with other members of a federation.

Tabelle 1: HLA-Regeln, nach Fujimoto 2000

Des Weiteren gibt es HLA-Regeln (Tabelle 1), die die Interaktion der Federates und Föderationen definiert. Das „Object Model Template“ (OMT) ist eine „Schablone“, die die Objekte und deren Wechselbeziehung genauer spezifiziert, wobei sich noch zwischen FOM (Federation Object Model) und SOM (Simulation Object Model) unterscheiden lässt. FOM definiert die gängigen Objekte, während SOM die Daten definiert (Fujimoto, 2000).

Die RTI bietet laut Straßburger und Klein (1997) sechs Schnittstellenfunktionalitäten an:

1) Federationmanagement:

Legt fest, wie die Federates Föderationen erzeugen oder ihnen beitreten können. Auch das allgemeine System wird definiert.

2) Deklarationsmanagement:

Hier wird definiert, wie die Federates der Teilsimulationen Informationen publizieren. Wenn ein Federate ein Objekt erzeugt oder ein von anderen Federates erstelltes Objekt nutzt, muss dies klar deklariert werden (Klein und Straßburger 1997).

3) Objektmanagement:

Diese Schnittstellenfunktionalität legt fest, wie die Federates Objekte nutzen können.

4) Eigentumsmanagement:

Dies erlaubt das Übergeben von Objektattributen. Auch kann immer nur ein Federate ein Objekt beeinflussen, sodass das System immer konsistent ist. Eine Interaktion setzt immer Eigentumsrechte von einem Federate voraus (Hintze et al., 2000).

5) Zeitmanagement:

Hier wird die Zeit in der Föderation und deren Einfluss auf die Objekte und die Interaktionen der Federates definiert.

6) Datenverteilungsmanagement:

Diese Schnittstellenfunktionalität definiert, wie die Datenübertragung zwischen den Federates abläuft.

Die HLA ist seit dem Jahr 2000 im „Institute of Electrical and Electronics Engineers“ als Standard für Modellierung und Simulation festgelegt.

2.3.2 CORBA

Nach Soliman (2000) lässt CORBA wie folgt beschreiben: CORBA ist die Abkürzung für „Common Object Request Broker Architecture“ und es handelt sich dabei um einen Standard für verteilte objektorientierte Anwendungen. CORBA wurde vor dem Hintergrund eingeführt, dass heterogene Anwendungen, welche also auf unterschiedlichen Programmiersprachen, Betriebssystemen oder Hardware basieren, miteinander kommunizieren sollen. Es handelt sich somit auch um eine Middleware, die für verteilte Simulation eingesetzt werden kann. Der ORB (Object Request Broker) ist dabei für Kommunikation der beteiligten Objekte verantwortlich und mit Hilfe der „Interface Definition Language“ (IDL) werden die Schnittstellen definiert, sodass auf Objekte zugegriffen werden kann. Jeden Aufruf eines CORBA Objekts vollzieht der ORB, sodass ein Zugriff auf Objekte und die Kommunikation in dem verteilten Netzwerk möglich ist, obwohl die einzelnen Komponenten heterogen sind.

2.3.3 GRIDS

Ähnlich wie die HLA ist GRIDS (Generic Runtime Infrastructure for Distributed Simulation) eine *Middleware*, die Kommunikation zwischen den Federates ermöglicht und Interoperabilität zwischen den Teilsimulationen herstellt (Taylor et al. 2003). Nach Taylor et.al (2003) bietet die HLA statische und festgelegte Funktionalitäten und im Vergleich dazu stellt GRIDS lediglich Basissimulationsservices, wie ein Simulations- oder ein Kommunikationsinterface, zu Verfügung, um so die Teilmodelle miteinander zu verbinden. GRIDS wurde 1997 definiert, um insbesondere komplexe Supply Chains mit Hilfe von verteilter Simulation zu realisieren. GRIDS kann, ähnlich wie die HLA, für eine Vielzahl von Simulationen angewendet werden (Taylor et al., 2003). GRIDS verfügt auch über ein „Object Exchange Model Template“ (OEMT), welches den Austausch von Objekten zwischen den Federates ermöglicht. Die vergleichbare HLA OMT fokussiert sich mehr auf den Informationsaustausch und bietet keine geeignete Möglichkeit, um die Objekte innerhalb der Föderation auszutauschen (Taylor et. al, 2003).

2.3.4 GRID Computing

Die Definition von GRID Computing ist sehr eng verwandt mit dem Begriff der verteilten Simulation und darf nicht mit GRIDS verwechselt werden. Bei verteilter Simulation handelt es sich um geographisch verteilte Computer, die zusammen *ein* komplexes Problem lösen. Im Vergleich dazu spricht man von GRID Computing, wenn sehr viele (Millionen) von Rechnern

weltweit zusammengeschlossen werden, um günstige Rechenkapazitäten für alle zu schaffen, die diese gerade benötigen. Hat ein Unternehmen also eine aufwändige Simulation und nicht die nötige Kapazität, können externe Ressourcen angefordert und genutzt werden. GRID Computing ist also nicht speziell für ein Problem ausgelegt (Parashar, Lee, 2005).

2.3.5 Multiagentensysteme

Diese Art der Simulation eignet sich besonders, um Systeme zu modellieren, die über autonome und heterogene Simulationsteilnehmer verfügen (Siebers, Aickelin, 2008). „Agent-Based Simulation“ (ABS) ist daher ein sehr realitätsnaher Simulationsansatz, denn vielfach unterscheiden sich die einzelnen Simulationsobjekte sehr. Das beste Beispiel ist die Simulation von Personen, denn diese handeln in der Regel in einer gewissen Umgebung unterschiedlich und jede einzelne Person verfügt über Stimmungen, Spontanität, Motivationen und viele weitere individuellen Eigenschaften, die sich schwer in einem Modell erfassen lassen. Nach Siebers und Aickelin (2008) sollte ein Agent über folgende Eigenschaften verfügen:

- Reaktionsfähigkeit: Ein Agent nimmt die Umgebung wahr und reagiert auf Änderungen
- Proaktives Verhalten: Ein Agent handelt zielgerichtet und opportunistisch
- Soziales Verhalten: Agenten interagieren miteinander
- Flexibilität: Agenten können ihre „Ziele“ auf unterschiedliche Weise erreichen

Ein Multiagentensystem (MAS) simuliert demzufolge die Interaktion vieler solcher autonomen Agenten und einige wissenschaftliche Beiträge befassen sich damit, wie verteilte Simulation für Multiagentensysteme angewendet werden kann. Hierbei ist der Beitrag „The Distributed Simulation of Multi-Agent Systems“ von Logan und Theodoropoulos (2001) zu nennen, der den Vorteil sieht, dass durch verteilte Simulationen die Multiagentensysteme immer komplexer werden können, da normale Rechner schnell an ihre Kapazitätsgrenzen stoßen. Die Projektarbeit wird zeigen, dass Multiagentensysteme ein viel beachtetes Forschungsgebiet im Zusammenhang mit verteilter Simulation darstellen.

2.4 Konzeptionelle Herangehensweise bei der Realisierung einer Simulation

Beim Modellieren gibt es eine logische Herangehensweise, um Systeme erfolgreich zu simulieren. Die im Folgenden erläuterten Schritte sind recht allgemein gehalten, allerdings sind sie relevant, um grundsätzlich zu verstehen, wie sich ein Entwickler einer Problemlösung nähert. Ausgehend von diesem Punkt konkretisiert sich die Denkweise von Entwicklern, sodass es leichter wird die Notwendigkeit von verteilter Simulation zu erkennen.

2.4.1 Konzept-Modellierung

Nach Robinson ist der Prozess der Konzept-Modellierung der wichtigste Schritt vor Erstellung einer Simulation und sollte Fragen wie die benötigten Daten, Ausführungszeit und Gültigkeit des Modells klären. Robinson (2004) behauptet, dass 50% einer erfolgreichen Simulation von einem guten Konzept-Modell abhängen, denn dieses hilft dabei das „Operational System“ zu verstehen und eine geeignete Simulation zu gestalten. Dieser Arbeitsschritt dient dazu real ablaufende Prozesse zu illustrieren, ohne dabei konkret etwas zu programmieren. Der Entwickler macht sich klar, in welcher Reihenfolge die Prozesse

ablaufen und wie sie sich beeinflussen. Heutzutage kann scheinbar eine Simulation ohne Konzept-Modell durchgeführt werden, indem man Softwarepakete nutzt. Diese verfügen in der Regel über vordefinierte Objekte und weitere anwenderfreundliche Werkzeuge, sodass die Erstellung des Konzept-Modells zunächst redundant erscheint. Allerdings muss bedacht werden, dass das Konzept-Modell unter anderem dafür genutzt wird das „Operational System“ besser zu verstehen, sodass dies weiterhin als grundlegender Schritt anzusehen ist. Vorteilhaft bei Softwarepaketen ist, dass nun schneller mit Experimenten begonnen werden kann, denn in den Softwarepaketen ist schon vieles vorgeben, was früher selbst hätte programmiert werden müssen. Robinson kritisiert auch, dass die Simulationen schnell zu komplex werden und Konzept-Modelle können dem entgegenwirken, indem irrelevante Faktoren schon zu Beginn nicht in ein Modell einbezogen werden.

Den Rahmen für das Konzept-Modell bilden nach Robinson (2004) diese vier Punkte:

1) Verständnis der Situation und des Problems

Um ein Problem adäquat zu lösen, wird zunächst eine einleuchtende Beschreibung der Situation und der Einflussfaktoren benötigt. Dieser Prozess wird zwischen dem Klienten und dem Entwickler durchgeführt, insbesondere Befragungen von Mitarbeitern spielen eine zentrale Rolle. Auch Schätzungen von nicht ermittelbaren Daten sollten diskutiert und festgesetzt werden.

2) Ziele des Modells

Dies ist der zentrale Punkt der Modellerstellung, da ausgehend von den Zielen experimentiert wird und sich die Ergebnisse daran messen lassen müssen. Dabei sollte berücksichtigt werden, was die Klienten erreichen wollen. Mögliche Ziele dabei wären Kostenreduktion oder ein schneller Produktionsablauf.

3) Inputs und Outputs

Man sollte sich im Klaren sein, dass bestimmte Faktoren gar nicht (Menge an Klienten in einem Restaurant) oder nur wenig beeinflusst werden können, während andere komplett in der Hand des Auftraggebers liegen (eingesetztes Personal). Alle diese Parameter sollten variiert werden, um so die reale Welt besser zu verstehen. Ziel dabei ist, dass das Modell visualisiert, warum sich eine Verbesserung eingestellt hat oder warum nicht. Beispielsweise sollten Engpässe erkennbar sein und wo diese auftreten.

4) Inhalt des Konzept-Modells

Der erste Schritt vor einer Simulation ist immer die Frage, ob eine Simulation der beste Weg ist um ein Problem zu lösen, denn Simulationen sind nur mit großem Arbeitsaufwand zu realisieren und Alternativen sollten in Erwägung gezogen werden. Danach wird der Inhalt des Modells festgelegt, um die Realität mit ausreichender Genauigkeit zu beschreiben. Auf der einen Seite ist abzuraten, ein zu detailliertes Konzept-Modell aufzustellen, weil dieses dann schnell unübersichtlich wird, allerdings muss gewährleistet sein, dass keine relevanten Punkte ausgelassen werden. Ein Modell für eine Produktionslinie sollte alle Fertigungsschritte beinhalten, während unerwartete und unregelmäßige auftretende Ereignisse hinsichtlich

ihrer Relevanz für das Simulationsmodell diskutiert werden sollten. Eine Mischung aus Kürze und Details ist erstrebenswert, denn irrelevante oder zu viele Faktoren machen das Modell schnell unübersichtlich.

2.4.2 Klassifizierung der Daten

Daten sind in Laufe einer Simulation einer der wichtigsten Faktoren, um ein Modell korrekt auszuführen. Normalerweise geht man bei dem Wort Daten von Zahlen aus, beispielsweise Zeitzyklen in einem Produktionsablauf oder durchschnittliche Ankunftszeiten, allerdings gibt es auch nicht numerische Daten, wie Beschreibung von Kundenverhalten, was sich äußerst schwer mit Zahlen quantifizieren lässt. Robinson (2004) teilt deshalb Daten in drei Klassen auf:

A) Kontextdaten

Auf diesen Daten beruht das ganze Simulationsmodell, denn diese sind verfügbar und lassen sich problemlos quantifizieren. Sie bilden die Grundlage und sollten deshalb möglichst genau sein. Beispiele wären: Anzahl von Maschinen und deren Arbeitsgänge; Fabrikgröße; Anzahl Angestellter.

B) Modelldaten

Diese Informationen sind wichtig für eine Simulation, allerdings sind sie zu Beginn noch nicht verfügbar. Diese Daten müssen deshalb gesammelt werden und lassen sich entweder empirisch ermitteln oder man führt Interviews mit Angestellten, die dann den konkreten Wert sehr realitätsnah schätzen. Beispiele wären: Fertigungszeit pro Produkt auf einer Maschine; durchschnittliche Zeit, die ein Angestellter für den Kundenservice braucht.

C) Daten, die das Modell verifizieren

Diese Werte sind weder verfügbar noch lassen sie sich bestimmen. In der Regel handelt sich um Vergleichsdaten aus der Realität, denn manchmal muss ein noch nicht existierendes System modelliert werden, was verlangt, dass alle Inputfaktoren geschätzt werden. Diese Vergleichsdaten dienen dann als Referenz, wobei immer die Sensitivität, das heißt die Toleranz gegenüber Abweichungen zwischen Realität und Schätzung, berücksichtigt werden muss.

Wie schon zu Beginn dieser Arbeit erwähnt, unterliegen fast alle Daten einer Unvorhersehbarkeit und Robinson (2004) stellt 4 Möglichkeiten vor, wie man dieses Problem löst.

1) „Traces“

Man sammelt alle Daten, die in dem realen System auftreten. Beispielsweise wird die exakte Servicezeit pro Kunde notiert und man fasst die Ergebnisse in einer Tabelle zusammen. Vorteilhaft an dieser Methode ist, dass die Daten gut und glaubhaft sind, allerdings die Variabilität leicht unterschätzt wird. Auch ist die Sensitivität kaum nachvollziehbar und man braucht in jedem Fall schon das reale System, um die Daten feststellen zu können.

2) Empirische Verteilungen

Tracedaten werden hier in einer Gruppe zusammengefasst, sodass man eine empirische Verteilung erhält. Man klassifiziert beispielsweise die Kundenbedienzeit (weniger als 1 Minute, 1-2 Minuten etc.) und hat so eine übersichtliche Darstellung der Informationen. Wenn man weiß, nach welchem Muster sich die Werte verteilen, kann problemlos durch Zufallszahlen die Stichprobe erweitert werden.

3) Statistische Verteilungen

Eine mathematische Funktion wird zugrunde gelegt, die dann zufällig die Werte generiert. Es gibt eine große Anzahl an Verteilungen (Normalverteilung, Erlangverteilung, Binomialverteilung, etc.), die unter Umständen sehr realitätsnahe Ergebnisse liefern. Vorteilhaft ist, dass so leicht eine große Stichprobe erzeugt und die Sensitivität ermittelt werden kann.

4) „Bootstrapping“

Dieses Verfahren eignet sich besonders dann, wenn man nur über eine kleine Stichprobe verfügt. Die vorher generierten „Tracedaten“, welche n Datensätze umfassen, werden zu Grunde gelegt und es wird eine Bootstrap-Stichprobe ermittelt, indem n -mal zufällig mit Zurücklegen gezogen wird. Demzufolge werden also aus der bestehenden Stichprobe neue Werte ermittelt, um so die Stichprobe zu erweitern.

2.4.3 Simulationsansätze

Heutzutage haben die meisten Simulationsprogramme eine graphische Oberfläche, sodass solche Simulation als VIS (Visual Interactive Simulation) beschrieben werden kann (Robinson, 2004). Diese Technologie wurde im Jahr 1976 zum ersten Mal von Hurion angewendet (Robinson, 2004) und vorteilhaft ist neben der Übersichtlichkeit auch, dass die Simulation gestoppt und Inputfaktoren geändert werden können. So kann der Entwickler direkt sehen, wie es zu Engpässen oder Leerständen kommt. Es wird somit leichter, das Modell zu verstehen und Verbesserungen zu erzielen. Im Allgemeinen gibt es nach Robinson (2004) drei verschiedene Möglichkeiten, Simulation computergesteuert durchzuführen. Erstens könnte man mit Hilfe von Tabellenkalkulationen simulieren, was sicherlich weniger bei komplexen Simulationen geeignet ist. Des Weiteren gibt es die Möglichkeit mit Programmiersprachen wie C++, Visual Basic oder Java eine Simulation zu erstellen, allerdings ist die gängigste Herangehensweise das Nutzen von Simulationssoftware. Diese bietet oft schon vordefinierte Objekte an, sodass der Programmieraufwand verringert wird. Zudem haben die Softwarepakete in der Regel eine graphische Oberfläche, was die Benutzerfreundlichkeit steigert. Kritisch an Standardsimulationssoftware ist, dass die Interoperabilität von unterschiedlichen Simulationen schwer oder unmöglich zu realisieren ist. Das Thema Interoperabilität und wie verteilte Simulation dabei helfen kann diese herzustellen, wird an anderer Stelle in dieser Projektarbeit thematisiert (siehe Kapitel 2.3 „Technischer Hintergrund“).

2.4.4 Erstellen eines Computermodells

Nachdem die Prozesse des Systems im Konzept-Modell visualisiert, die Daten gesammelt und eine Simulationsart festgelegt wurde, ist der nächste Schritt das Schreiben des Codes in der entsprechenden Software. Robinson (2004) hebt hervor, dass neben der Zeit, die man zum Schreiben des Codes braucht, auch Transparenz, Flexibilität und Ausführungsgeschwindigkeit wichtig sind. Transparenz gibt Auskunft darüber, wie schnell ein Code von anderen Personen verstanden werden kann, während Flexibilität die Möglichkeit beschreibt, einen Code anzupassen. Die Ausführungsgeschwindigkeit ist besonders bei aufwändigen Programmen wichtig, denn Rechenleistung ist begrenzt. Diese Problematik ist insbesondere im Zusammenhang mit der verteilten Simulation ein wichtiges Thema. Die konzeptionelle Herangehensweise dient dazu, möglichst effizient Simulationen zu erstellen. Gute und anpassbare Modelle können dann beispielsweise bei der Erzeugung einer Gesamtsimulation eingebunden werden, sodass auch eine gute Gesamtsimulation erstellt werden kann. Ist dies nicht der Fall, würde die Qualität der Gesamtsimulation deutlich abnehmen.

2.5 Herausforderungen der verteilten Simulation

2.5.1 Synchronisation von Prozessen

In diesem Abschnitt werden Herausforderungen bei der technischen Umsetzung von verteilter Simulation dargestellt. Es wird ein System betrachtet, in dem unterschiedliche Prozesse ablaufen. Diese real ablaufenden Prozesse modelliert man durch logische Prozesse, die zeitlich klar definierten Ereignissequenzen entsprechen (Fujimoto, 2000). Da in einem System mehrere logische Prozesse (LP) ablaufen, interagieren diese miteinander und erzeugen neue Ereignisse in einem anderen Prozess. Fujimoto erläutert dies anhand des Startvorgangs eines Flugzeuges, denn wenn es abhebt (LP in dem System Flughafen A), würde so direkt ein Landeereignis in einem anderen Flughafen B erzeugt werden. Man könnte nun meinen, dass verteilte Simulation optimal wäre, um solche verknüpften Ereignisketten zu simulieren, da jedem LP ein Prozessor zugeordnet werden könnte. Diese würden dann Prozesse simulieren und neue Ereignisse in den anderen LP terminieren. Das Problem hierbei ist, dass alle Ereignisse, auch diejenigen, die durch andere logische Prozesse entstanden sind, in jedem LP in korrekter zeitlicher Abfolge berechnet werden müssen. Es kann vorkommen, dass Ereignisse nicht in ihrer richtigen zeitlichen Abfolge simuliert werden, sodass ein logischer Prozess Ereignisse in der Vergangenheit eines anderen logischen Prozesses erzeugt. Sobald die LP auf verschiedenen Prozessoren ablaufen, treten schneller logische Fehler in der Ereignisliste auf. Diese werden als „Causality Errors“ (Fujimoto, 2000) bezeichnet. „Synchronization Problem“ bezeichnet das Problem, bei welchem die Ereignisse in falscher zeitlicher Abfolge simuliert werden. Es gibt Lösungsansätze, die logischen Fehler zu vermeiden, allerdings bringen diese auch ihre Schwierigkeiten mit sich. Ein Ansatz ist, dass erst Berechnungen durchgeführt werden, wenn es sicher ist, dass von einem anderen logischen Prozess kein neues Ereignis generiert wird. Es wird eine „first-in-first-out queue“ (Fujimoto, 2000) erstellt, die garantiert, dass alle Ereignisse in der richtigen Reihenfolge bearbeitet werden. Dieser Ansatz kann genutzt werden, solange sich mindestens eine

Nachricht in dieser Schlange befindet. Ist dies nicht der Fall, kann es allerdings zu einem „Deadlock“ führen, was bedeutet, dass jeder logische Prozess auf mögliche Ereignisse eines anderen logischen Prozesses wartet. Das System ist nun blockiert. Das lässt sich am besten an Hand des Beispiels von Fujimoto erläutern:

Es gibt drei Flughäfen: Los Angeles, New York und Chicago

Es ist bekannt, dass jedes neue Ereignis, das seinen Ursprung in Chicago hat, frühestens zum Zeitpunkt $t=5$ in New York relevant sein wird. Bei Ereignissen mit dem Ursprung in Los Angeles wäre es sogar $t=9$.

Man betrachtet nun die Ereignisreihenfolge für den New Yorker Flughafen mit folgenden Zeitpunkten:

T	Ursprungsort
2	Los Angeles
4	Chicago
5	Chicago
8	Los Angeles
9	Los Angeles

Tabelle 2: ankommende Flüge in Flughafensimulation

Die Simulation stoppt hier nach dem dritten Ereignis (zum Zeitpunkt $t=5$), obwohl schon für $t=8$ ein weiteres Ereignis terminiert wurde, das simuliert werden könnte. Allerdings ist nicht gewährleistet, dass vor $t=8$ andere Ereignisse relevant werden. Dies kommt daher, dass lediglich definiert ist, dass neue Ereignisse ausgehend von Chicago mindestens zum Zeitpunkt $t=5$ eintreten. Es ist also nicht sicher, dass zum Zeitpunkt 6 oder 7 ein neues Ereignis simuliert werden muss. Somit stoppt die Simulation und wenn dies auf allen parallel arbeitenden Prozessoren passiert, spricht man von einem Deadlock. Gelöst kann dieses Problem mit „Lookaheadnachrichten“, welche besagen, dass in einer bestimmten Periode kein neues Ereignis eintrifft. In dem Flughafenbeispiel wäre dies die Flugzeit (hier 3 Stunden). Hat der Flughafen in New York zum Zeitpunkt $t=5$ noch keine Nachricht der anderen Flughäfen erhalten, könnte er zum Zeitpunkt $t=8$ springen, da in dieser Zeit keine Ladungen möglich sind, da es bei $t=5$ keine Starts gegeben hat. Die Nachricht trüge dann lediglich die Information, dass vor einem bestimmten Zeitpunkt definitiv kein neues Ereignis eintreten wird.

Der Lookaheadwert ist ein sehr wichtiger Aspekt der verteilten Simulation, wenn die Simulationsperformance betrachtet wird. Je kleiner der Lookaheadwert ist, desto kleinschrittiger wird die Simulation berechnet, was zu inakzeptablen Simulationszeiten führen kann. Die Auswertung der wissenschaftlichen Beiträge wird zeigen, dass der Lookaheadwert bei Performanceuntersuchungen eine zentrale Rolle spielt.

2.5.2 Zeitkonsistenz

Verteilte Simulation kann genutzt werden um viele individuelle Teilsimulationen zu realisieren und die Zeitinkonsistenz spielt dabei eine wichtige Rolle. Die Teilsimulationen müssen nicht zwangsläufig einen parallel ablaufenden Zeitfortschritt haben, allerdings sollte, wie im

vorangegangenen Abschnitt beschrieben, zeitliche Inkonsistenz nicht zu logischen Fehlern führen. Im Fall von virtuellen Trainingsumgebungen ist die zeitliche Inkonsistenz allerdings eines der Kernprobleme (Schumann 2004). Diese tritt beispielsweise auf, wenn in einem Szenario mehrere Teilnehmer direkt miteinander kommunizieren und deren Interaktionen Einfluss aufeinander haben. Eine durchgeführte Aktion benötigt eine Kommunikationszeit, sodass eine durchgeführte Zustandsänderung von einem Teilnehmer erst mit einer kleineren Verzögerung für die anderen Simulationsteilnehmer sichtbar wird. Um die zeitliche Konsistenz wieder herzustellen, aktualisiert sich das Szenario erst immer nach Ablauf der Rundlaufzeit, sodass alle Teilnehmer zum selben Zeitpunkt die Änderungen im Szenario wahrnehmen können. Dies ermöglicht die Konsistenz der Simulationen, allerdings beeinträchtigt es die Interaktivität der Computersimulation (Schumann, 2004). Wird eine virtuelle Umgebung mit Hilfe einer verteilten Simulation realisiert, kann die HLA das oben beschriebene Problem angehen. Ein zentraler Bestandteil der HLA-Architektur ist das Objektmanagement, das nur einem Federate die Möglichkeit gewährt, die Objektattribute zu ändern (Schumann 2004). Somit können nicht, möglicherweise zeitlicher Inkonsistenz geschuldet, zwei Simulationsteilnehmer auf ein Objekt zugreifen und deren Zustand ändern. Zudem ist in der HLA ein Zeitmanagement verankert, welches „für die Synchronisation der Federates benutzt werden“ (Schumann 2004) kann. Somit lässt sich grundsätzlich die zeitliche Inkonsistenz von virtuellen Umgebungen angehen, was den Vorteil von einheitlichen Standards, wie der HLA, zeigt.

Kapitel 3: Anwendungspotenzial verteilter Simulation

Wie bereits in dieser Projektarbeit erwähnt, wurde die verteilte Simulation und die damit einhergehende Implementierung der HLA Ende der 90er Jahre aktuell. Die Publikation „Zivile Anwendungspotential der High Level Architecture (HLA)“ der Wissenschaftler Klein und Straßburger wagt in dem Jahr 1997 einen Ausblick auf die möglichen Anwendungen dieser Technologie. Als großes Anwendungsgebiet prognostizieren Straßburger und Klein zunächst den Verkehr sowie die Logistik. Hierbei nennen sie die Simulation von Verkehrsnetzen, Transportflotten und der gesamten Infrastruktur als potenziellen Anwendungsbereich. Insbesondere die urbanen Räume bieten viele Bereiche, wie den privaten und öffentlichen Personenverkehr, die auch schon zu der damaligen Zeit über den großen Vorteil verfügen, dass umfangreiche (Echtzeit)-Daten vorhanden sind.

Weitergehend sehen sie die Produktion, Fertigung und Fertigungsplanung als mögliche Anwendungsbereiche, da diese bereits über zahlreiche Simulationen und Modelle verfügen, allerdings diese nur selten kombiniert werden, weil sie geographisch getrennt sind. Die HLA bietet dabei die geeignete Architektur, um komplexe Gesamtsimulationen als Kombination der bereits vorhandenen Teilsimulationen zu erstellen.

Die dritte Anwendungsmöglichkeit sehen Klein und Straßburger im Notfall- und Katastrophenmanagement („Emergency Management“), denn auch hier fallen viele relevanten Daten, mit großer Sicherheit geographisch verteilt, an, sodass verteilte Simulation einen adäquaten Lösungsansatz darstellt.

Im folgenden Teil der Projektarbeit werden zunächst Entwicklungen und Potenziale in dem Bereich der verteilten Simulation beschrieben und zudem werden einige der Anwendungen vorgestellt. Die zugrunde liegende Quellen sind vornehmlich die fachbezogenen Tagungen wie die ASIM und die Winter Simulation Conference. Das „Journal of Simulation“ oder die Beiträge der Fachtagung „SIGSIM PADS“ bieten zudem noch weitere nützliche Forschungsergebnisse zu diesem Thema.

3.1 Erste Forschung für zivile Anwendungen

In der Fachtagung der ASIM im Jahr 2000 war ein zentrales Thema das Überführen von Daten und Erkenntnissen von einer Simulation in eine andere. Einer der ersten Realisierungsansätze war, dass eine Programmiersprache entwickelt werden sollte, mit der dann neutrale Modelle erstellt werden können (Rabe, 2000). Diese sollten auch in jedes beliebige Simulationsmodell übertragbar sein, was die Wiederverwendbarkeit und Nutzbarkeit erhöht hätte. Diese Idee erwies sich zu dem damaligen Zeitpunkt als nicht praxisfähig, da auch spezielles Systemverhalten oder Regeln in einer Simulation in der neutralen Sprache geschrieben wurde. Da keines der in der Praxis genutzten Simulationsprogramme diese Sprache verstehen konnte, weil diese neu war, musste jeder Simulationsprovider ein Übersetzungsprogramm entwerfen, sodass die Simulationssprache konvertiert werden kann. Somit wurde lediglich eine neue Sprache entwickelt, deren Übersetzung für jedes Simulationsprogramm erwies sich allerdings nicht als umsetzbar. Es entstand die Idee, dass nicht versucht wird eine Simulation in eine andere zu übertragen, sondern vielmehr die einzelnen Simulationen in ihren

Simulationsprogrammen zu belassen und lediglich Interoperabilität zwischen heterogener Simulationssoftware hergestellt werden soll (Rabe, 2000). Die verteilte Simulation und insbesondere der in der Entwicklung stehende Standard der HLA boten Potenzial zur Lösung dieses Problems. Die Idee, die verteilte Simulation für zivile Anwendungen zu nutzen, wurde nun erforscht. Ende der 90er Jahre wurden mehr die zivilen Anwendungsbereiche betrachtet, da zu diesem Zeitpunkt verteilte Simulation fast ausschließlich für militärische Anwendungen, wie virtuelle Trainingsumgebungen, genutzt wurde.

Bedingungen für die verteilte Simulation sind laut Rabe (2000):

- Kommunikationsmechanismus:
Eine Kommunikation zwischen den Simulationsprogrammen muss möglich sein, insbesondere dann, wenn Ereignisse auf mehrere Teilsimulationen einen Einfluss haben.
- Zeitsynchronisation:
Die Zeit in den Simulationsprogrammen muss aufeinander abgestimmt werden.
- Eindeutige Objektattribute:
Informationen über Objektattribute müssen von allen Teilsimulationen unterstützt und verstanden werden.

Zudem wurden einige Herausforderungen festgestellt, denn es muss eine Methode gefunden werden, um die komplexen Verbindungen zwischen den internen Objekten einer Simulation und „Domainobjekten“ (auf diese wird in allen Teilsimulationen zugegriffen) zu realisieren. Da es sehr unwahrscheinlich ist, dass alle Unternehmen auf dieselbe Software zurückgreifen, um ihre Simulationen durchzuführen, wurde das MISSION Project („Modelling and Simulation Environments for Design, Planning and Operation of Globally Distributed Enterprises“) ins Leben gerufen. Dieses sollte einen übergeordneten Standard festlegen, in dem beispielsweise grundlegende Objekte oder Schnittstellen verschiedener Simulationen definiert sind. Es sollten also übergeordnete „Tools“ implementiert werden, die in einer Vielzahl von Prozessen der Teilsimulationen angewendet werden. Somit kann die Anwenderfreundlichkeit gesteigert werden, denn durch eine benutzerfreundliche Infrastruktur von Simulationsmodellen wird vieles schon im Voraus bereitgestellt, sodass ein Nutzer nicht zwangsläufig viel selbst programmieren muss. Beispielsweise könnten direkt nutzbare Standardobjekte in der „Template Library“ für die Nutzer bereitgestellt werden (Rabe, 2000). Die Forschung im MISSION Project begann Ende der 90er Jahre und im November 2001 wurde der Abschlussbericht veröffentlicht (online einsehbar unter: www.ims.org). Historisch war das MISSION Project bedeutend, wobei in dieser Projektarbeit die wissenschaftlichen Ergebnisse unmittelbar danach betrachtet werden.

Im Folgenden werden konkrete Anwendungen der verteilten Simulation beschrieben. Zunächst wird auf Basis von Lendermann (2006) verdeutlicht, welche potentiellen Anwendungsgebiete vor knapp 10 Jahren in Betracht gezogen wurden und welche Vorteile verteilte Simulation für bestimmte Szenarien hat. Die im Folgenden beschriebenen Beispiele dienen zur Orientierung und zeigen Ideen sowie konkrete Anwendungen in verschiedenen

Bereichen, die in dem Zeitraum 2000 bis 2015 entwickelt wurden. Danach folgt eine Einschätzung des Zukunftspotenzials der verteilten Simulation.

3.2 Analysen der potentiellen Anwendungsgebiete in der Logistik

Verteilte Simulation hat schon seit vielen Jahren das Problem, dass diese Technologie oft trotz vieler potentieller ziviler Anwendungsgebiete nicht genutzt wird, sondern nur sehr starke Anwendung für militärische Zwecke hat (Lendermann, 2006). Im Folgenden wird nun betrachtet, welche Gründe es gibt, dass verteilte Simulation in der Produktion und der Logistik wenig genutzt wird. Bedingt durch die Globalisierung hat man heutzutage vielfach Systeme, die so komplex sind, dass die verteilte Simulation ideal wäre, um die Probleme effektiv zu lösen. In komplexen Supply Chains könnte verteilte Simulation für die einzelnen Unternehmen eine sehr gute Option sein, relevante Information in einer Simulation zusammenzuführen, ohne dabei alle Details für die anderen Unternehmen offenzulegen. Das ist insbesondere vor dem Hintergrund von Urheberschutz interessant für viele Unternehmen. Die folgenden Abschnitte sollen zeigen, für welche Bereiche verteilte Simulation sehr gut geeignet ist und seine Anwendung findet oder ob das Anwendungspotenzial wenig praxisfähig ist.

3.2.1 Vorteile verteilter Simulation gegenüber normaler Simulationssoftware

Zu Beginn dieser Arbeit wurde schon eingehend beschrieben, warum Simulationen wichtig sind und welche Vorteile diese für jedes Unternehmen haben. Durch die erwähnte Auslegung und Programmierung wird allerdings deutlich, wo die Schwäche von „normaler“ Simulationssoftware liegt. Nachdem das zu analysierende System verstanden und in eine Simulation übertragen wurde, beginnt man die Parameter zu variieren, um das computergesteuerte Modell so möglichst realitätsnah zu gestalten. Die Simulation beruht auf statistischen Verteilungen und durch mehrfaches Wiederholen wird versucht, die Realität darzustellen, um so Ergebnisse für strategische Entscheidungen des Unternehmens zu gewinnen. Das Ziel ist also, das aktuelle System zu verstehen, um so Änderungen einschätzen zu können. Langfristig sollen die Änderungen dann zu einem verbesserten Ablauf in einem Unternehmen führen. Genau hier sieht Lendermann (2006) das Potenzial für die verteilte Simulation, denn durch verkürzte Produktlebenszyklen, ändernde Kundenanforderungen und steigende Produktvielfalt wird es immer schwieriger, einen „Steady-State“ in der Produktion und der Logistik zu erhalten (Lendermann, 2006). Der „Steady-State“ wäre zum Beispiel ein nicht schwankender Output einer Produktion und diesen zu optimieren ist erstrebenswert, wenn viele Faktoren planbar sind und nur in einem geringen Maße schwanken. Nach Lendermann (2006) ist dies aber immer seltener der Fall. Simulationen, die langfristig Prozesse in der Produktion und der Logistik optimieren wollen, können eventuell in der Realität gar nicht angewendet werden, da sich die Einflüsse für das Unternehmen zu schnell ändern. Man hat vielmehr ein Interesse daran, die dynamischen Änderungen in einer Simulation zu berücksichtigen, insbesondere dann, wenn etwas Unvorhergesehenes passiert. Neue Parameter könnten eine gute Simulation unbrauchbar machen, sodass eine schnelle, auf

simulationsgestützten Erkenntnissen basierende Entscheidung nicht getroffen werden kann. Im Folgenden wird nun zunächst dargestellt, welchen Problemen verteilte Simulation gegenübersteht und für welche Anwendungen verteilte Simulation eine adäquate Lösung darstellen kann.

Ergänzend dazu hat verteilte Simulation den weiteren Vorteil, dass dasselbe Simulationsmodell zeitgleich auf mehreren Rechnern ausgeführt werden kann, sodass sich lediglich die zugrunde liegenden Parameter unterscheiden würden (Raab, Schulze, Straßburger, 2008b). So können schneller unterschiedliche Szenarien simuliert und gute Lösungen für das reale System generiert werden.

3.2.2 Herausforderungen bei praktischer Anwendung von verteilter Simulation

In Kapitel 1.5 wurden bereits technische Herausforderungen von verteilter Simulation, wie die Zeitkonsistenz und die Synchronisation von Prozessen erwähnt. Neben den technischen Problemen muss auch bei der praktischen Umsetzung von verteilter Simulation einiges beachtet werden. Lendermann (2006) stellt dabei drei Kernprobleme fest:

- 1) Anpassungsfähigkeit verteilter Simulation bei Änderung der Einflussfaktoren
- 2) Modellierungszeit
- 3) Ausführungsgeschwindigkeit

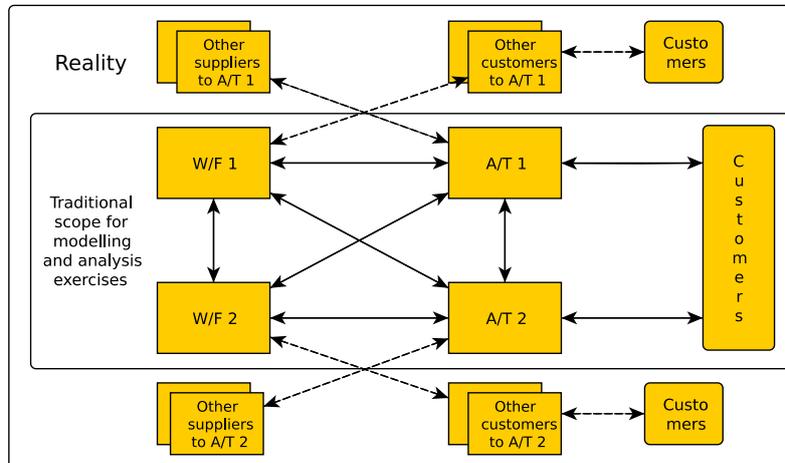


Abbildung 4: Supply Chain Netzwerk, nach Lendermann 2006

Das erste Problem kann am besten anhand eines Beispiels verdeutlicht werden. Die dargestellte Abbildung 4 zeigt vereinfacht eine Supply Chain einer Halbleiterfabrik. Es entspricht nicht der Realität, wenn eine Simulation nur eine Fabrik (W/F: Wafer Fab) und ein A/T (Assembly and Testing) modellieren würde, denn

Supply Chains sind in der Regel weitaus komplexer. Der eingerahmte Bereich in der Grafik wäre trotzdem simulierbar, allerdings ist auch dieses Modell nicht genügend. Viele externe Unternehmen beeinflussen die Supply Chain und in diesem Fall sind es zusätzliche Lieferanten und Konsumenten. Diese in die Simulation einzubinden ist eine Herausforderung, da die neuen Faktoren alle heterogen sind (Lendermann, 2006). Des Weiteren ist immer ein Problem von Simulationsprogrammen, dass Entscheidungen von Menschen vorhergesagt werden

müssen. Entscheidungen von Personen sind zwar in einem gewissen Maße vorhersehbar, allerdings fallen diese auch immer wieder unabhängige Entscheidungen.

Die Modellierungsgeschwindigkeit ist eine Herausforderung, denn es ist immer Zeit notwendig, um Änderungen in einem Modell zu berücksichtigen. Insbesondere sollten Anpassungen des Modells so schnell umgesetzt werden, dass die gewonnenen Erkenntnisse der Simulation in der Realität auch noch umgesetzt werden können, allerdings ist dies nach Lendermann (2006) nicht immer der Fall. Das ebenfalls von dem Autor identifizierte Problem der Ausführgeschwindigkeit, was beispielsweise mit der Wahl des Lookaheadwertes zusammenhängt, ist auch eine Herausforderung der verteilten Simulation. In der in Kapitel 4 zu findenden Analyse ist zu erkennen, dass die „Performanceanalyse“ von verteilter Simulation ein beachtetes Forschungsgebiet ist, denn diese kann die Simulationszeit durch die vielen Prozessoren beträchtlich reduzieren, allerdings kommt es auch oft zu ungenügend langen Ausführgeschwindigkeiten. Die technischen Gründe werden in wissenschaftlichen Beiträgen, die in Kapitel 4 erwähnt werden, intensiv diskutiert, allerdings würde eine exakte Beschreibung den Rahmen dieser Projektarbeit überschreiten.

Eine weitere Herausforderung für die verteilte Simulation ist die Interoperabilität von Standardsimulationssoftwarepaketen. Im Optimalfall könnten heterogene Simulationsmodelle quasi mittels „Plug-and-Play“ direkt miteinander kombiniert werden, was allerdings noch nicht möglich ist (Raab, Schulze, Straßburger 2008a). Ergänzend dazu nennen die Autoren eine weitere Herausforderung, denn syntaktische Interoperabilität könne zwar mit Hilfe der HLA gewährleistet werden, allerdings fehle bislang etwas Vergleichbares für die semantische Korrektheit.

3.3 Anwendungen verteilter Simulation

In diesem Abschnitt werden nun einige konkrete Beispiele genannt, in denen verteilte Simulation genutzt wurde, um reale Systeme zu modellieren. Dies soll zeigen, dass verteilte Simulation nicht nur theoretisch genutzt werden kann, sondern es in den letzten 15 Jahren auch zahlreiche konkrete Anwendungen gab. Neben diesen konkreten Anwendungen werden auch einige potenzielle Szenarien genannt, die prädestiniert wären für einen verteilten Simulationsansatz.

3.3.1 Anwendungsbeispiele Logistik

Eines der typischen logistischen Probleme ist die Tourenplanung von Speditionen, welche heutzutage vielfach mit Hilfe von linearer Programmierung berechnet wird, indem die Kostenfunktionen minimiert werden (Juan et al. 2010). Juan et al. präsentieren einen anderen Ansatz, der mit verteilter Simulation realisiert wird. Normale lineare Programmierung kann zum Beispiel mit der „Clarke-White Savings Methode“, welche *eine* optimale Route zwischen einem Depot und seinen n Klienten ausgibt, liefern, indem Wegersparnisse betrachtet werden. Das Problem ist hier allerdings, dass es immer nur *eine* optimale Lösung gibt und Juan et al. (2010) nutzen ein Multiagentensystem um „pseudo-optimale Lösungen“ zu generieren,

sodass der Kunde je nach Präferenz aus verschiedenen Lösungen auswählen kann. Dies wird realisiert, indem eine Zufallsfunktion implementiert wird, sodass nicht immer die größte Ersparnis ausgewählt wird, sondern dass lediglich die Wahrscheinlichkeit am größten ist, dass diese im Optimierungsprozess ausgewählt wird. Somit variiert die resultierende Lösung des Problems.

Der „Multiagenten Ansatz“ besagt, dass der vorher erläuterte Lösungsansatz parallel auf vielen Prozessoren durchgeführt wird, sodass der Algorithmus zur selben Zeit mehrmals ausgeführt wird. Diese parallelen Simulationsausführung bezeichnet man als „Agenten“, denn diese „suchen“ alle individuell nach der besten (pseudo-)optimalen Lösung. Dabei findet unter den „Agenten“ eine Kommunikation statt, sodass schneller der optimale Lösungsweg gefunden wird.

Juan et al. (2010) führen zudem noch einen Performancetest durch, welcher überprüft, wie schnell eine Simulation mit n Prozessoren, die jeweils t Agenten erzeugen, innerhalb von 10 Sekunden eine Lösung finden. Die Ergebnisse zeigen, dass eine Maschine mit 10 Threads schon eine gute Lösung präsentiert (lediglich 30% Abweichung von der optimalen Lösung). 250 Agenten (25 Prozessoren mit je 10 Threads) liefern innerhalb von 10 Sekunden nur eine 10%ige Abweichung von der optimalen Lösung und dieser Wert kann durch das Hinzufügen weiterer Kerne kaum verbessert werden.

Dieses Beispiel zeigt, dass verteilte Simulation genutzt werden kann, um ein klassisches Probleme wie eine Tourenplanung anders zu lösen. Der klassische Ansatz mit Hilfe von linearer Programmierung liefert auch sehr gute Ergebnisse für solche Probleme, allerdings steigert verteilte Simulation mit Hilfe der genutzten Agenten die Anzahl der Lösungen und somit können differenzierte und individuelle Entscheidungen getroffen werden. Der Kunde kann nach anderen Entscheidungskriterien eine Lösung auswählen, was die Flexibilität erhöht. Weiter kann verteilte Simulation für andere logistische Probleme genutzt werden.

Eines wäre die Organisation der Lastwagen in Häfen, um einen schnellen Umschlag von den

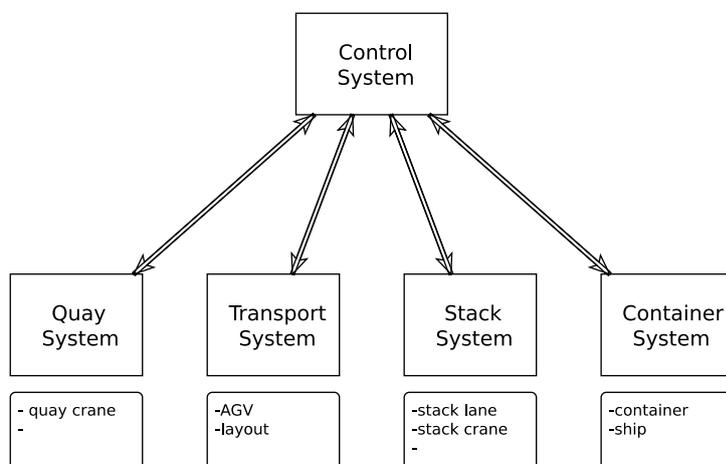


Abbildung 5: Hauptmodelle der Hafensimulation, nach Duinkerken 2002

Waren zu gewährleisten (Lendermann, 2006). Es gibt Simulationen, die verschiedene Anordnungen der Container prüfen oder die Wege der Lastwagen simulieren, allerdings müssten alle diese Teilsimulationen kombiniert werden, um eine optimale Lösungen für dieses Problem zu finden. Lendermann (2006) sieht aus diesem Grund Potenzial für diese Anwendung und

Duinkerken et al. (2002) liefern eine konkrete Lösung für die Modellierung eines Hafens unter Zuhilfenahme der verteilten Simulation. Die Abbildung 5 zeigt die vier Hauptsysteme, die die jeweiligen Teilsimulationen darstellen, jeweils mit deren wichtigen Objekten. Duinkerken et al. (2002) nutzen also verteilte Simulation anders als Juan et al., denn hier wird konkret gezeigt, dass eine Kombination von Teilsimulationen vorteilhaft gegenüber einem monolithischen Gesamtmodell sein kann.

Ein weiteres Szenario, in dem verteilte Simulation nützlich sein kann und erfolgreich umgesetzt wurde, wird von Kalasky et al. (2010) beschrieben. In diesem Beitrag geht es um die TSA (Transportation Security Administration), welche die Sicherheit von 450 Flughäfen gewährleisten soll. Die TSA muss den Personalbedarf an jedem Flughafen individuell planen, denn dieser unterliegt kontinuierlichen Schwankungen je nach Wochentag und Tageszeit. Es soll nun ein neues und effizienteres Simulationsmodell entwickelt werden, denn zuvor wurde mit Hilfe von zwei Simulationsprogrammen das Personal verwaltet. Eine Simulation berechnete im 5-Minuten-Takt den Personalbedarf für jeden Einsatzort, beispielsweise an der Sicherheitskontrolle für die Passagiere. Das zweite Simulationsmodell bestimmt den optimalen Einsatzplan, also die Arbeitszeiten, von den verfügbaren Voll- und Teilzeitkräften. Eines der Ziele war nun, diese beiden Modelle zu koppeln, sowie möglichst viele Simulationen zeitgleich ablaufen zu lassen, denn vorher konnte immer nur eine Simulation mit beträchtlicher Simulationszeit durchgeführt werden. Der Beitrag von Kalasky hat gezeigt, dass die technische Realisierung möglich war und soll hier nicht im Detail erläutert werden, sondern es sollen vielmehr die Ideen für Anwendungsszenarien vermittelt werden.

Auch Lendermann (2006) präsentiert eine mögliche Anwendung verteilter Simulation für logistische Anwendungen, denn er betrachtete die Ersatzteillogistik von Flugzeugen. Technische Probleme von Flugzeugen, die durch Verschleißteile ausgelöst werden, sind eine große Herausforderung für Fluggesellschaften, da dies hohe Kosten durch Verspätungen nach sich ziehen kann. Normalerweise werden die Ersatzteile direkt an die Fluggesellschaften verkauft, die dann vorsichtshalber an dem Zielflughafen diese wichtigen Teile vorbereiten, allerdings wird eine andere Herangehensweise vorgeschlagen. Die wichtigen Ersatzteile sollten vielmehr beim OEM (Original Equipment Manufacturer) verbleiben und dieser gewährleistet dann die Wartungsarbeiten an den jeweiligen Flughäfen. Simulationen könnten dann ermöglichen dieses dynamische System zu kontrollieren. Die Mindestbestände müssen immer überwacht sowie aufgefüllt und Lieferzeiten müssen kalkuliert werden. Zudem müssen Flugräume analysiert werden, sodass die Ersatzteile immer über kurze Distanzen zwischen den Flughäfen transportiert werden, falls in einem Ersatzteillager Bedarf besteht (Lendermann, 2006). Es handelt sich also um ein komplexes System, da zum einen viele Daten anfallen und Teilsimulationen kombiniert werden sollen, beispielsweise spielen die Flugpläne der einzelnen Flughäfen eine Rolle für die Ersatzteillager am Boden.

3.3.2 Anwendungsbeispiele in der Produktion

Auch hier stellt Lendermann (2006) ein interessantes Szenario vor, in dem verteilte Simulation, hier in dem Kontext von industrieller Produktion, genutzt werden kann. Die Idee

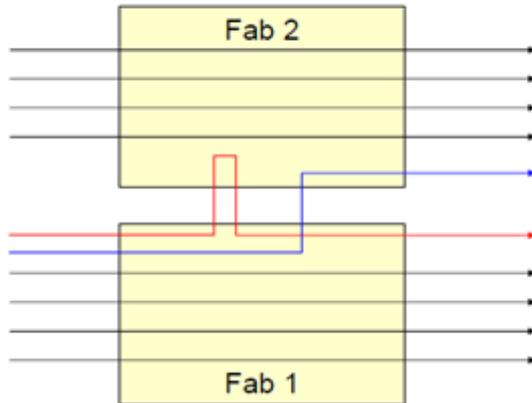
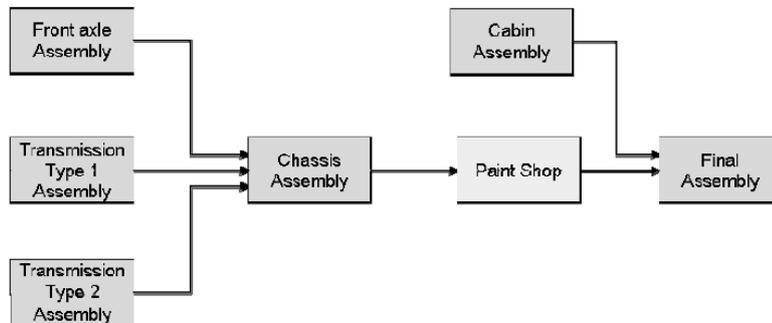


Abbildung 6: „Borderless Fab“, Lendermann 2006

wird als „Borderless Fab“, also grenzenloses Fabrik bezeichnet und wird in Abbildung 6 gezeigt. Dieser Ansatz wäre insbesondere für Unternehmen relevant, die mehrere Fabriken in räumlicher Nähe betreiben und einer wechselnden Nachfrage ausgesetzt sind. Im Fall eines unvorhergesehenen Fehlers oder einer Störung einer Fabrik könnte man Kapazitäten einer anderen Fabrik nutzen, um so den Produktionszyklus nicht zu sehr zu verlängern. Insbesondere dann, wenn die Fabriken

unterschiedlichen Unternehmen gehören, wäre verteilte Simulation ein geeigneter Ansatz, um Produktionsressourcen gemeinsam zu nutzen. Falls Firmen zusammen an einem gemeinsamen Projekt arbeiten, könnte die Produktion von verschiedenen Firmen so besser aufeinander angestimmt werden, ohne dass alle relevanten Daten für eine gemeinsame Simulation offengelegt werden müssen.

Ein weiteres Beispiel für die Anwendung von verteilter Simulation in der Produktion wird von Raab et al. (2008a) dargestellt. Die



nebenstehende Übersicht 7 zeigt die sieben Sektionen in einer Automobilfabrik, für die jeweils ein

Abbildung 7: Materialfluss in einer Automobilfabrik, Raab et al.2008a

eigenes Simulationsmodell erstellt wurde. Durch den Materialfluss innerhalb der Fabrik sind die Sektionen miteinander verbunden und beeinflussen sich also gegenseitig. Das Ziel war, diese Teilsimulation in einem Gesamtmodell zu vereinen und insbesondere die Verbindungen zwischen den Sektionen adäquat darzustellen. Die Abbildung 8 zeigt, wie Verbindung der Teilmodelle mit Hilfe der HLA realisiert wurde. Auch hier wird auf die konkrete Beschreibung der technischen Umsetzung verzichtet, denn es soll lediglich gezeigt werden, warum verteilte Simulation auch in der Produktion großen Nutzen bringen könnte.

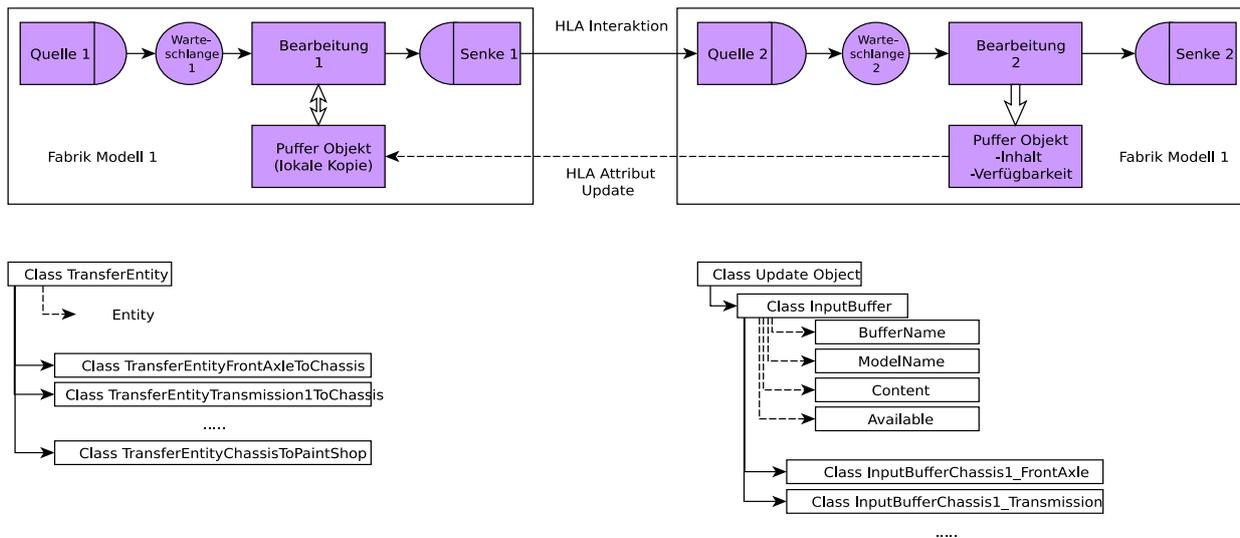


Abbildung 8: HLA-basierte Interaktion von Simulationsmodellen, nach Raab et al. 2008a

Eine Anwendung von verteilter Simulation wurde von auch von Daniluk und ten Hompel (2015) dargestellt. In ihrem Beitrag geht es um eine multiagentenbasierte Materialflusssteuerung in einer Fabrik. Dabei werden die Bearbeitungsstationen und Verbindungen, also die einzelnen Materialflussmodule, abbildet und diese agieren als Agenten. Die Arbeitsgegenstände werden mit Hilfe von RFID-Chips, welche beispielsweise die Zielinformationen tragen, von den einzelnen Materialflussmodulen gelesen. Auf Basis von drei unterschiedlichen Simulationsumgebungen (716, 1663 und 3074 Agenten), 28 Modellklassen (Simulationsmodell mit entsprechenden Eigenschaften) wurde auf 5 Rechnern für die verteilte Simulation die Modelle getestet. Die Ergebnisse sind sehr unterschiedlich und zeigen, dass es zu einem deutlich Speedup durch verteilte Simulation kommen kann, aber nur unter bestimmten Umständen. Teilweise wurde sogar eine sehr starke Verlangsamung festgestellt, die sich mit einem geringen Lookaheadwert erklären lässt. Im Optimalfall wurde eine Verbesserung um den Faktor 3,2 gegenüber der Ausführung auf nur einem Rechner erreicht. Das Problem ist, dass der Kommunikationsaufwand wegen der steigenden Anzahl der Teilmodelle deutlich zunimmt, was die Simulationsgeschwindigkeit herabsetzt. Umso komplexer das simulierte System, größer der Lookaheadwert und mehr Elemente das System umfasst, desto besser eignet sich die verteilte Simulation, da sich ein Speedup der Simulationsgeschwindigkeit erwarten lässt (Daniluk, ten Hompel 2015).

Dieses Beispiel zeigt, in welchem Maße verteilte Simulation zu einem Performancegewinn führen kann, allerdings nicht zwangsläufig führen muss. Im letzten Kapitel der Projektarbeit wird etwas detaillierter auf die Performanceuntersuchung im Zusammenhang mit verteilter Simulation eingegangen.

3.3.3 Verkehrssimulationen

Hier soll eine konkrete Anwendung beschrieben werden, die sich mit einem schon über Jahre aktuellen Thema beschäftigt und zwar der Verkehrsplanung und Steuerung in Großstädten. Eine effiziente Verkehrsplanung könnte Abgasbelastungen minimieren, Staus reduzieren und somit die Lebensqualität erhöhen. Suzumura und Kanezashi (2014) zeigen am Beispiel von der Stadt Dublin, wie verteilte Simulation in diesem Szenario genutzt werden kann. Das Kernanliegen der beiden Forscher ist, dass alle anfallenden Daten der Verkehrsmittel, also die Informationen zu den privaten PKW, Bussen, Zügen und U-Bahnen, in eine Simulation einfließen. So sollen bessere Prognosen für die Verkehrsplanung der Stadt Dublin generiert werden.

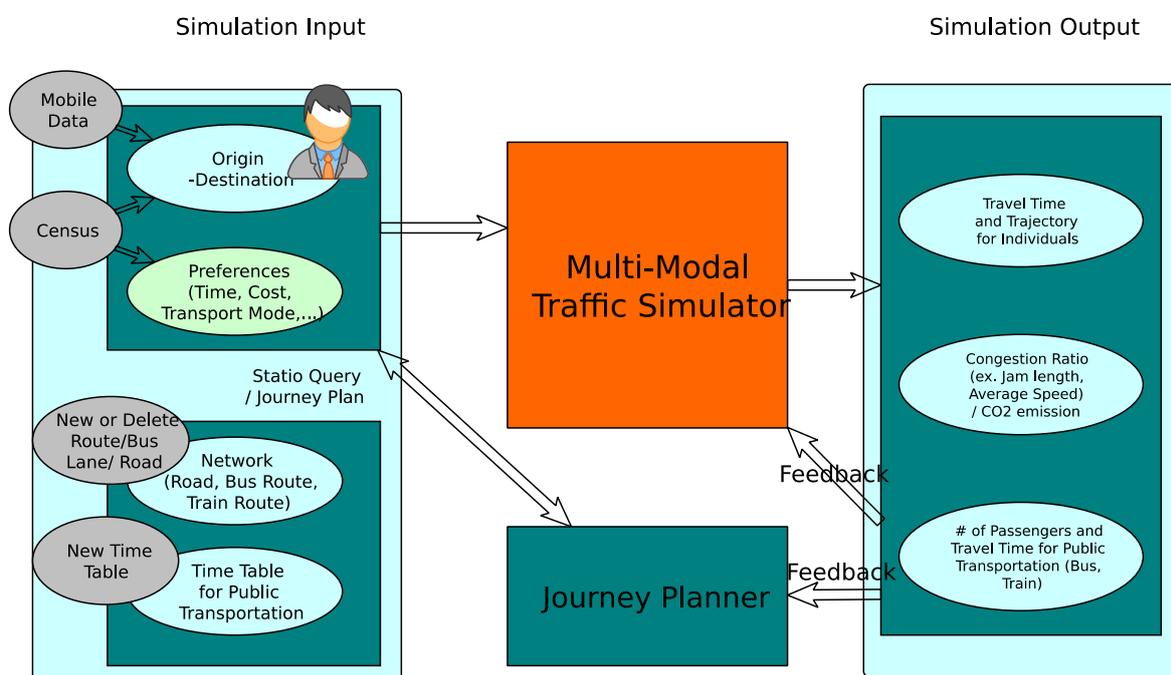


Abbildung 9: Komponenten der Verkehrssimulation, nach Suzumura und Kanezashi 2014

Abbildung 9 zeigt insbesondere, welche Daten als „Input“ für die Simulation gesammelt und umgesetzt werden müssen. Der „Journey Planner“ generiert die individuellen Routen der Personen. Dies geschieht auf Basis der Präferenzen, denn manche Personen fahren nur mit dem Auto oder nur mit öffentlichen Verkehrsmitteln. Zudem werden Start-Ziel-Informationen und auch die Zeitpläne von Bussen und Bahnen in Echtzeit in der Simulation berücksichtigt. So kann die schnellste individuelle Route errechnet und zudem weitere Analysen durchgeführt werden (CO₂ Ausstoß oder Staulänge). Charakteristisch für dieses Problem ist, dass die Daten zum einen sehr umfangreich sind und zum anderen an vielen verschiedenen Stellen anfallen, sodass man von einer verteilten Simulationsumgebung spricht. Diese Simulation wurde mit Hilfe der XAXIS (X10-based Agent eXecutive Infrastructure for Simulation), dies ist eine Multiagentenplattform, realisiert, welche eine Middleware besitzt, die es erlaubt, die Simulation auch für „large-scale environments“ (Suzumura, Kanezashi 2014) auszuführen.

Dies bedeutet, dass mit Hilfe von vielen und leistungsstarken Prozessoren die Simulationsgeschwindigkeit signifikant erhöht werden soll. Am Beispiel von der Stadt Dublin wird ein Performancetest durchgeführt, der die Verbesserung misst, die man durch zusätzliche Rechenkern erreicht. Diese Untersuchung soll nun kurz geschildert werden, damit verdeutlicht wird, in welchem Maße zusätzliche Prozessoren die Simulationszeit herabsetzen.

Betrachtet wird zunächst die Situation mit einem Node (12 Threads), mit dem man innerhalb von 140 Sekunden 100 Simulationsschritte berechnen kann (Großraum Dublin). Um die Simulationsschritte für das Zentrum von Dublin durchzuführen, würde man nur 70 Sekunden benötigen. Dies lässt sich mit der geringeren Datenmenge begründen. Ein Simulationsschritt bildet ca. 100 Sekunden ab, sodass in 140 Sekunden Rechenzeit 2 Stunden und 50 Minuten Echtzeit simuliert werden kann. Der Großraum Dublins umfasst 60,000 Verkehrsknotenpunkte und das Zentrum lediglich 17,000. Nun wird das Szenario mit mehreren Rechenkernen durchgeführt. Die Ergebnisse zeigen, dass die Performance um den Faktor 2,3 verbessert werden kann, allerdings tritt eine „Sättigung des Performancegewinns“ ein. Mit steigender Prozessorenanzahl sinkt logischerweise die Arbeitsauslastung je Prozessor, sodass die Performance je Prozessor negativ beeinflusst oder sogar zu gering ist, damit der Prozessor arbeiten kann. In einem anderen Szenario wird die Anzahl der simulierten Fahrten und die Anzahl der Verkehrsknotenpunkte deutlich erhöht, sodass selbst bei 12 eingesetzten Kernen jeder genügend Arbeitsbelastung ausgesetzt ist. Die Messergebnisse zeigen, dass der Performancegewinn (12 Prozessoren im Vergleich zu einem Kern) ca. bei dem Faktor 5 liegt. Somit zeigen Suzumura und Kanezashi, dass eine signifikant schnellere Rechenleistung durch verteilte Simulation erzielt werden kann, allerdings muss die Anzahl der Prozessoren jeweils dem konkreten Rechenumfang angepasst werden.

3.3.4 Virtuelle Trainingsumgebungen

Dobermann und Stelbe (2000) präsentieren ein mögliches Anwendungsszenario von verteilter Simulation in der Schiffsindustrie, wobei hier der Fokus auf Aus- und Weiterbildung des Schiffspersonals liegt. Bestimmte Problemsituationen, wie eine Havarie, sollen trainiert werden. Charakteristisch für die maritime Trainingsumgebung ist, dass es viele verschiedene Simulationsbereiche gibt, wie die Navigation des Schiffes, die Maschinen oder auch die Interaktionen der Schiffe untereinander. Um die Gesamtsimulation praxisnäher zu gestalten, sollen die erwähnten Teilsimulationen aufeinander Einfluss haben und deshalb wurde die verteilte Simulation genutzt.

Die Simulation soll so gestaltet sein, dass sie je nach Teilnehmerzahl erweitert werden kann. All dies geschieht auf Basis eines einheitlichen „Simulations-Framework“ (Dobermann, Stelbe 2000). Zum Beispiel beschreiben Dobermann und Stelbe (2000) die Entwicklung eines speziellen Kartenservers, der die Schifffavigation simulieren soll. Die relevanten Informationen der Seegebiete sind mittlerweile standardisiert (ECDIS: Electronic Chart Display and Information Systems), wobei die Daten noch klassifiziert werden können. Beispielsweise wäre es möglich, dass Kunden je nach ihren Wünschen die zu Grunde liegende Karte weiter spezifizieren, indem sie zusätzliche Informationen, wie die Tiefe des Meeres in dem jeweiligen

Seegebiet hinzufügen. Diese Daten werden insbesondere dann wichtig, wenn die Simulation Teilnehmer enthält, die autonom handeln. Diese könnten theoretisch von den gängigen Schiffstraßen abweichen und dann muss das System über weitere Daten verfügen, damit es realistisch ist. Mit Hilfe der HLA können in diesem Trainingszenario gut dynamische Daten, wie die Position von Schiffen, umgesetzt werden (Dobermann, Stelbe 2000).

Eine nützliche Anwendung ist der Schiffsimulator. Auf Basis der Karteninformation kann eine Person die wichtigen Parameter, wie Geschwindigkeit oder Kurs, beeinflussen und steuert so ein Schiff wie in einer Kommandobrücke. Dies kann sehr gut mit der Simulation der Maschinen verknüpft werden, denn diese könnten direkt die Befehle der Brücke umsetzen. Zudem gibt es eine Trainerkomponente, mit der die vollständige Simulation überwacht wird oder auch eingegriffen werden kann.

3.4 Zukunftspotenziale verteilter Simulation

Nachdem einige konkrete Beispiele der verteilten Simulation präsentiert wurden, sollen nun, basierend auf einer Umfrage von Straßburger und Schulze (2008) aus dem Jahr 2008, Anwendungsgebiete dieses Simulationsansatzes eingeschätzt werden. Die beiden Autoren untersuchten die Zukunftspotenziale. Hierzu wurden 61 Personen aus verschiedenen Industrie- und Forschungsbereichen befragt und die Abbildung 10 zeigt die Ergebnisse. Auffällig ist, dass den Anwendungen 2,3 und 4 eine recht hohe Relevanz zugewiesen wird und das ist vermutlich darauf zurückzuführen, dass hier die Vorteile der verteilten Simulation zur Geltung kommen.

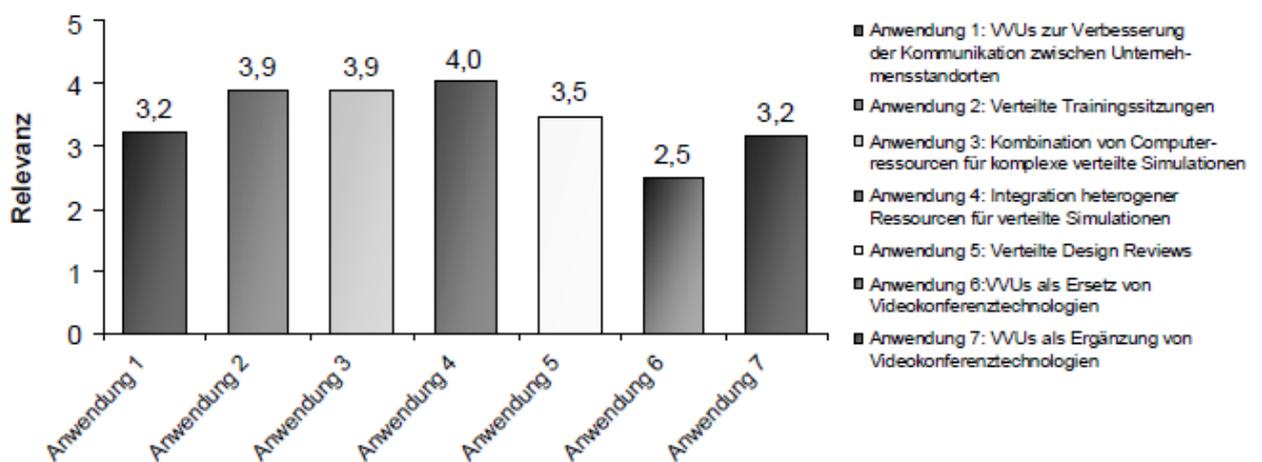


Abbildung 10: Anwendungspotenzial verteilter Simulation, Straßburger und Schulze 2008

Insbesondere Anwendung 4, welche besagt, dass heterogene Ressourcen integriert werden können, ist eine Idee, die bereits 1997 von Straßburger und Klein als eine Motivation für die Forschung in diesem Bereich genannt wurde.

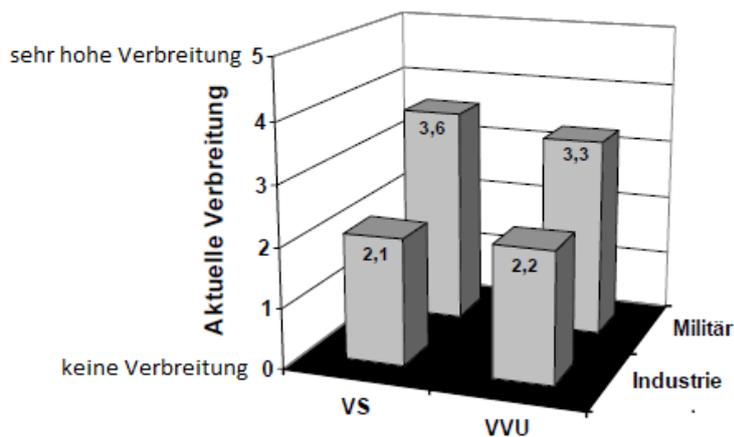


Abbildung 11: Verbreitungsgrad verteilter Simulation, nach Straßburger Schulze 2008

Bei der Befragung der Teilnehmer bezüglich der Anwendungsbereiche wurden Integrationsaspekte, Produktion und Produktionsentwicklung genannt, was sich nur geringfügig vom den Erwartungen aus dem Jahr 1997 unterscheidet.

Ein neuer Aspekt, der 2008 von den Teilnehmern der Studie häufiger genannt wurde, ist der „Know-How-Schutz“

(Straßburger, Schulze 2008), da beispielsweise Simulationen genutzt werden sollen, damit Unternehmen gemeinsam an der Entwicklung eines Produktes mitwirken können. Mitwirkende Firmen wollen allerdings vielfach nicht konkrete Details ihrer Produktkomponenten offenlegen. Eine weitere interessante Statistik (Abbildung 11) zeigt den Verbreitungsgrad der verteilten Simulation sowie der verteilten virtuellen Umgebung für militärische Zwecke und die Industrie. Im Vergleich zum Militär, durch das die verteilte Simulation seinen Anfang genommen hat, ist die Verbreitung in der Industrie noch auf keinem hohen Niveau. Dies lässt sich auf „keinen zufriedenstellenden Reifegrad“ (Straßburger, Schulze 2008) dieser Technologie zurückführen. Die Entwicklungen der HLA werden im Jahr 2008 als „ausgereift“ (Straßburger und Schulze, 2008) angesehen, allerdings werden auch hier Schwächen ausgemacht. Die Anwendung ist zum einem schwierig und zeitaufwändig und zum anderen wird durch die HLA lediglich der Syntax der Simulationen aufeinander abgestimmt und die semantische Korrektheit nicht ausreichend berücksichtigt. Dies alles führt dazu, dass verteilte Simulation, insbesondere im Vergleich zu militärischen Anwendungen, keine zufriedenstellende Verbreitung für zivile Anwendungen hat.

Nachdem nun die zugrunde liegende Technik von verteilter Simulation, deren Anwendungen sowie das Zukunftspotenzial dargestellt wurden, sollen in dem nun folgenden Kapitel die wissenschaftliche Beiträge zum Thema verteilter Simulation kategorisiert werden. Ziel ist es Entwicklungspfade darzustellen und die Hauptforschungsgebiete zu identifizieren. Auch werden die Themen hinsichtlich ihrer Relevanz, Chancen und Nutzen für zivile Anwendungsbereiche eingeschätzt.

Kapitel 4: Entwicklungspfade und Kategorisierung der Forschung zur verteilten Simulation

In diesem Abschnitt der Projektarbeit sollen die wissenschaftlichen Beiträge aus den letzten Jahren gegliedert und eingeordnet werden. Wie schon bei den oben erwähnten Beispielen zu erkennen ist, sind die Forschung und die Anwendungsbereiche der verteilten Simulation sehr vielfältig, sodass es eine Herausforderung ist, die Beiträge zu kategorisieren. Die wichtigen wissenschaftlichen Arbeiten werden in der Regel in diesen Medien publiziert beziehungsweise in deren Konferenzen präsentiert:

- Fachtagungen der ASIM (Arbeitsgruppe Simulation)
- Journal of Simulation
- SIGSIM PADS (SIGSIM Conference on Principles of Advanced Discrete Simulation)
- Winter Simulation Conference (WSC)

Bei der durchgeführten Gliederung wurden nur die Publikationen der Fachtagung ASIM sowie der Winterconference einbezogen. Im „Journal of Simulation“ werden wissenschaftliche Beiträge zum Thema Simulation veröffentlicht, allerdings gibt es nur sehr wenige Beiträge speziell zu der verteilten Simulation. Im August 2012 wurde eine Ausgabe komplett „Parallel and Distributed Simulation“ gewidmet, allerdings waren diese Beiträge wenig hilfreich für die in dieser Arbeit vorgenommene Analyse. Der Grund war, dass die Publikationen sich mehr mit der IT-Infrastruktur und der technischen Umsetzung beschäftigen, allerdings steht in dieser Einordnung mehr der Nutzen der verteilten Simulation für konkrete Anwendungen in Wirtschaft im Vordergrund. Die drei Fachtagungen beschäftigen sich alle mit dem Thema Simulation und es konnten fast in jedem Jahr Beiträge speziell zur verteilten Simulation gefunden werden. Die Winterconference und SIGSIM PADS finden jeweils jährlich statt und die ASIM wird im 2-jährigen Turnus abgehalten. Somit kann gut die Relevanz dieses Themas über einen längeren Zeitraum verglichen werden. Für diese Analyse wurden Beiträge der ASIM sowie der Winter Simulation Conference betrachtet. Da erst seit Ende der 90er Jahre verteilte Simulation wissenschaftlich betrachtet wurde, bilden die Beiträge im Zeitraum von 2000 bis zum Jahr 2015 die Grundlage für die Analyse:

	2000	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	Σ
WSC	7	7	10	13	9	13	14	9	9	5	8	7	5	6	7	0	129
ASIM	7	- *	- **	- *	3	- *	- **	- *	3	- *	4	- *	- *	0	- *	1	18

Tabelle 3: Anzahl der wissenschaftlichen Publikationen zum Thema verteilter Simulation

*: es fand keine ASIM Fachtagung statt

** in diesen Jahren konnte ich auf keine Beiträge zugreifen

Die obige Tabelle zeigt die berücksichtigten Publikationen für Kategorisierung der verteilten Simulation. In Summe wurden 147 Publikationen betrachtet und dies ist einer der Gründe, warum diese Projektarbeit diese zwei Fachtagungen als Grundlage nimmt. Zum einen bilden 147 Texte schon eine aussagekräftige und repräsentative Grundlage, um Entwicklungspfade einzuschätzen und verschiedene Entwicklungen zu präsentieren und zum anderen konnte

nicht auf alle Volltexte der SIGSIM PADS zugegriffen werden, sodass sich die Einbindung dieser Arbeiten komplizierter gestaltet. Durch das Hinzunehmen aller Publikationen zum Thema verteilter Simulation der SIDSIM PADS wäre die Anzahl um eine beträchtliche Anzahl an Beiträgen gewachsen. Schon auf Basis der 147 Beiträge gestaltet sich eine Analyse durch die Anwendungsvielfalt schwierig und weitere Beiträge hätten diesen Effekt noch verstärkt. Auch ist zu erwarten, dass Beiträge in ähnlicher Form in beiden Fachtagungen verwendet wurden. Aufgrund der genannten Probleme der SIGSIM PADS werden nun im Folgenden die Publikationen der ASIM sowie der Winter Simulation Conference in differenzierter Form betrachtet.

4.1 Herausforderungen bei der Datenanalyse

Aufgrund der Menge und der inhaltlichen Beschaffenheit der Publikationen ist es schwierig, Oberbegriffe zu definieren und anhand dieser dann alle Beiträge zu kategorisieren. Grob könnte man diese Einteilung wählen:

- 1) Anwendungen: Logistik, Produktion, virtuelle Umgebungen, etc.
- 2) Technische Untersuchung der verteilten Simulation: Performance von verteilter Simulation, IT Infrastruktur, Interoperabilität von Teilsimulation, HLA

Diese grobe Einteilung in Anwendungen und technische Untersuchung erwies sich nicht als geeignet, da Beiträge in der Regel mehreren Bereichen zugeordnet werden können. Ein hypothetischer Beitrag könnten virtuelle Umgebungen nutzen, um Prozesse in der Produktion darzustellen. Um dies zu realisieren müsste zudem an der IT-Infrastruktur etwas geändert werden, um schließlich die Interoperabilität zu gewährleisten. Dies zeigt, dass man je nach Blickwinkel einen Beitrag vielen verschiedenen Kategorien zuordnen kann, sodass bei einer willkürlichen Zuordnung die Gliederung mangelhaft wäre. Die nebenstehende Abbildung visualisiert schon eher die Einteilung der Publikationen. Man liest dieses Schaubild von Innen,

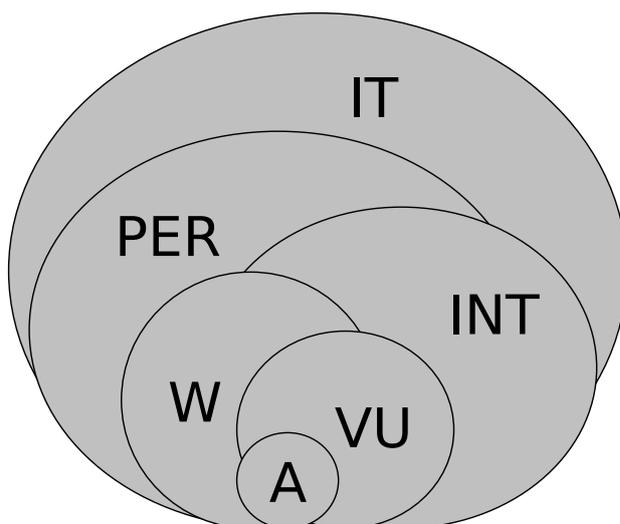


Abbildung 12: Themenübersicht der analysierten wissenschaftlichen Beiträge

also den Anwendungen (A) nach außen zur IT (IT-Architektur). Die Anwendungen (A) umfassen die Logistik, Produktion, Verkehrssimulationen, Militär sowie Fabrik und Serviceplanung. Virtuelle Umgebungen (VU) ist die nächste „Schale“, denn es gibt auch genügend Beiträge, die sich speziell auf virtuelle Umgebungen beziehen, allerdings kann diese Information auch für konkrete Anwendungen genutzt werden. Weiteres (W) umfasst zunächst sonstige Anwendungen, Umfragen oder Beiträge, die beispielsweise das ökonomische Potenzial von verteilter Simulation betrachten. Es handelt sich also um interdisziplinäre Beiträge. Der

Bereich „INT“ befasst sich mit der Interoperabilität von Teilsimulationen, was mehr einen

technischen Hintergrund hat, allerdings auf lange Sicht sich für konkrete Anwendungen nutzen lassen kann. Ebenso technisch ist der Abschnitt der Performance (PER), denn diese Beiträge diskutieren beispielsweise Ansätze zur Herabsetzung der Rechenzeit. „Umschlossen“ werden alle diese Teilbereiche von der IT-Infrastruktur (IT) und Publikationen zu diesem Thema befassen sich mit der grundlegenden Technik. Beispielsweise wird allgemein die HLA betrachtet oder Alternativen dazu vorgeschlagen. Des Weiteren würden Beiträge zu neuen Algorithmen oder des MISSION Projekts in diese Kategorie fallen.

Durch die angesprochene Vielfalt der Inhalte wurden zahlreiche Beiträge mehreren dieser Bereiche zugeordnet, sodass die Summe der jedem Teilbereich zugeordneten Beiträge nicht

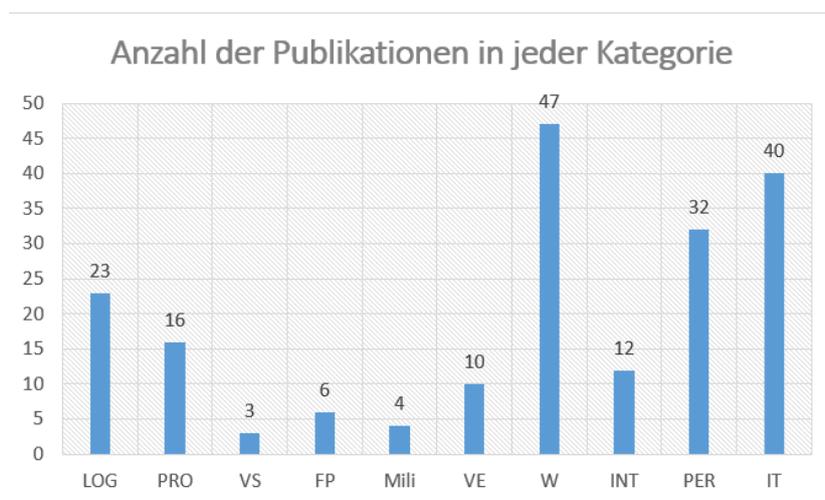


Abbildung 13: Quantitative Zuordnung der Publikationen in ihre thematischen Kategorien

Legende: siehe Abkürzungsverzeichnis

der Gesamtsumme entspricht. Hintergrund für diese Einteilung ist, dass man prinzipiell den Entwicklungsstand eines Forschungsgebietes ablesen kann. Umso mehr Anwendungen es gibt, desto ausgereifter ist die zu Grunde liegende Technik. Je mehr Publikationen sich mit Grundlagen, wie der HLA befassen, desto mehr muss an der Basis von verteilter Simulation geforscht werden.

4.2 Gliederungen der Forschungsgebiete

Im Folgenden werden die einzelnen Kategorien der verteilten Simulation etwas detaillierter erläutert. Die in Abbildung 12 visualisierten Anwendungen (A) werden zu Beginn beschrieben, wobei hier noch in die Teilbereiche Logistik, Produktion, Verkehrssteuerung, Service- und Fabrikplanung und Militär unterteilt wird. Ausgehend davon werden andere Aspekte, wie die Interoperabilität oder Performanceanalyse mit in die Analyse eingebunden, was in einer Auswertung am Schluss dieser Projektarbeit resultiert. Während der Analyse der Beiträge werden aus Gründen der Übersichtlichkeit keine Artikelnamen genannt, sondern es werden nur im Klammern stehende Referenznummern angegeben. Die Referenznummern beziehen sich auf den Anhang, in dem die exakten Artikelnamen und Autoren genannt werden.

4.2.1 Logistische Anwendungen

Die Auswertung hat ergeben, dass unter den 147 betrachteten Beiträgen 23 einen *konkreten* logistischen Hintergrund haben. Die Betonung liegt hier auf konkret, weil viele Forschungsergebnisse dafür genutzt werden können, dass auch Probleme der Logistik gelöst werden. Publikationen dieser Rubrik beschreiben also recht spezifische Fragestellungen der Logistik und geben konkrete Beispiele für Anwendungen der verteilten Simulation. Aufgrund

der immer komplexer werdenden Supply Chain ist die Anwendung der verteilten Simulation hierfür nachvollziehbar, da vielfach schon vorhandene Simulation gekoppelt werden müssen. Die wissenschaftlichen Arbeiten befassen sich unter anderem hiermit:

- 1) Tourenplanungen 146
- 2) Zeit- und Arbeitskräfteplanung 13, 104
- 3) Materialflüsse in Unternehmen 143, 147
- 4) Anwendungsbeispiele 23, 73, 118, 122, 144
- 5) Interoperabilität von COTS für logistische Anwendungen 14, 24, 77

Wie man unschwer erkennen kann, ist auch der Oberbegriff „Logistik“ unzureichend, da sich die Logistik in zahlreichen Unterkategorien aufteilt. Im Folgenden sollen nun einige Entwicklungen und Beispiele genannt werden, sodass man einen Überblick über die verteilte Simulation in der Logistik erhält.

Einige konkrete Anwendungen wurden in den Publikationen präsentiert, wie zum Beispiel die Simulation eines Containerterminals (23). Alternativ wurde verteilte Simulationsumgebungen genutzt, um Katastrophenhilfe leisten zu können, indem effektiver Rettungskräfte eingesetzt werden (118). Auch gab es ein Beispiel von einem Pharmakonzern, der Materialflüsse in einem Lager mit Hilfe von verteilter Simulation plant und umsetzt (144). Etwas allgemeiner wurden Anwendungen wie Tourenplanung oder Zeit- und Arbeitskräfteplanung betrachtet. All diese Anwendungen zeigen, dass logistische Probleme, insbesondere bedingt durch sehr differenzierte Supply Chains und intralogistische Abläufe, verteilte Simulation Potenzial hat und deshalb wurde im Vergleich zu den anderen Themen recht viel zu diesen Themen publiziert. Zum einen kann also verteilte Simulation dabei helfen bereits bestehende Systeme, die schon in geeigneten Teilsimulationsprogrammen realisiert wurden, zu koppeln. Zum anderen kann verteilte Simulation auch noch nicht umgesetzte Planungen modellieren, wie das Beispiel des „virtuellen Unternehmens“ zeigt (12). Unter virtuellen Unternehmen versteht man in diesem Zusammenhang, dass Wissen und Kapazitäten von Unternehmen geteilt werden, um so Aufträge auszuführen, die normalerweise, bedingt durch den hohen Arbeitsaufwand, nicht realisierbar wären (Venkateswaran et al., 2001). Zudem thematisieren einige Beiträge speziell Probleme und Lösungen bei Interoperabilität von logistischen Teilsimulationen (14, 24, 77). Zusammenfassend ist zu sagen, dass logistischen Anwendungen das größte Potenzial zugemessen wird, da zum einen quantitativ mehr publiziert wurde und zum anderen deutlich mehr allgemeine Szenarien beschrieben werden, in denen komplexe Systeme, etwa eine Supply Chain, mit Hilfe von verteilter Simulation besser simuliert werden kann. Die in Kapitel 3 erwähnten Beispiele stehen exemplarisch für die Vorteile, denn es können heterogene Simulationen gekoppelt werden (91) und effiziente Zeitpläne für komplexe logistische Systeme erstellt werden (122).

4.2.2 Verteilte Simulation in der Produktion

Diesem Anwendungsfeld konnten 16 Beiträge zugeordnet werden, wobei nur wenige Publikationen konkrete Beispiele für verteilte Simulation nannten. Es stand vielmehr im Vordergrund, wie verteilte Simulation in der Produktion genutzt werden kann und welche Potenziale diese Art der Simulation hat. Hier sind zum Beispiel die Publikationen zum MISSION

Projekt zu nennen (5,6) oder es gibt Ansätze, komplexe Fertigungsprozesse mit Hilfe von einem Multiagentensystem zu realisieren (138). Neben konkreten Anwendungen (91: gekoppelte Simulation eines Krans und eines Montageplatzes) oder Ansätzen, die die Zeitplanung in der Produktion verbessern sollen (43), stehen hier vielmehr grundlegende Ideen zur technischen Verbesserung im Vordergrund. Es ist aber auch festzuhalten, dass im Vergleich zu den anderen Anwendungsbereichen zwar quantitativ viel vorhanden ist, allerdings wesentlich weniger konkrete Anwendungen erwähnt werden.

Potenzial liegt besonders darin, dass heutzutage oft ein Produkt von verschiedenen Unternehmen zusammen produziert wird, allerdings wollen die Firmen nicht ihre gesamten Informationen teilen. Auch kann, wie in den Anwendungsbeispielen in Kapitel 2 erwähnt, verteilte Simulation genutzt werden, um Produktionskapazitäten über Unternehmensgrenzen hinaus zu nutzen. Diese zwei Aspekte stellen ein großes Potenzial dar, denn es werden auf der einen Seite interne Informationen geschützt und auf der anderen Seite kann Produktion individueller und in Kombination mit anderen Unternehmen gestaltet werden.

Ähnlich wie bei den logistischen Anwendungen, wurden bisher vielfach Simulationen genutzt, die lediglich Abschnitte einer Produktionskette darstellen. Beiträge wie „Management of HLA-Based Distributed Legacy SLX-Models“ (87) identifizieren diese Herausforderungen und Lösungen zur Interoperabilität von verschiedenen Simulationen.

4.2.3 Verkehrssteuerung

In diese Kategorie fallen genau drei Beiträge, die auch sehr praxisnah sind. Eine Publikation beschreibt eine konkrete Anwendung, denn es soll die Verkehrssteuerung der Stadt Dublin verbessern, indem mit Hilfe der verteilten Simulation alle relevanten Daten in einer Simulation umgesetzt werden (123). Diese Publikation wurde in dem vorgegangenen Kapitel ausführlich präsentiert. Ein Beitrag der ASIM 2000 beschreibt die Kombination aus zwei Simulationsprogrammen, denn es soll ein Fahrsimulator mit einer Simulation gekoppelt werden, die möglichst realitätsnah den Verkehr darstellt (132). Somit muss eine Person einen virtuellen PKW steuern und steht zudem in direkter Interaktion mit einer fiktiven, aber realitätsnahen Verkehrssituation. Ergänzt wird diese Kategorie durch eine Publikation, die beschreibt wie mit Hilfe von Echtzeitdaten und einem Multiagentensystem Verkehrsteilnehmer geleitet werden, um so möglichst zeiteffizient am Ziel anzukommen (101).

Auch hier hat verteilte Simulation ein gewisses Potenzial, allerdings ist dieser Aspekt, insbesondere im Vergleich für Anwendung in der Logistik, unterrepräsentiert. Dies ist auf der einen Seite damit zu erklären, dass der Begriff Logistik wesentlich allgemeiner ist, sodass logischerweise mehr Beiträge diesem Thema zugeordnet werden können. Durch die Komplexität von Verkehrssimulationen bietet sich dieses Thema eigentlich an, diese spezielle Simulationsform anzuwenden, wobei der Forschungsschwerpunkt mehr auf logistischen und produktionstechnischen Anwendungen zu liegen scheint.

4.2.4 Fabrik- und Serviceplanung

Diese Kategorie kann als Ergänzung zu 4.2.2 angesehen werden, da sich diese beiden Bereiche thematisch sehr ähnlich sind. Der Unterschied ist, dass sich das Unterkapitel 4.2.2 vielmehr damit befasst, wie bestehende Simulationsmodelle von Produktionssystemen mit Hilfe der verteilten Simulation optimiert werden. Im Vergleich dazu liefert dieser Abschnitt Referenzen zu wissenschaftlichen Beiträgen, die sich mit den Arbeitsschritten vor der Produktion befassen.

Die Fabrik- und die Serviceplanung bietet auch ein interessantes Forschungsgebiet zur verteilten Simulation, denn auf der einen Seite sind heutzutage oft viele Teilnehmer in eine der Planung eines neuen Services oder Produktes involviert, allerdings wollen diese auch ihre Daten schützen. Insbesondere die Beiträge 102, 96 und 36 setzen sich damit auseinander, wie über Unternehmensgrenzen hinaus Planungen auf Basis von Simulationen durchgeführt werden können. Ergänzend dazu gibt es Ideen, wie das „Borderless Fab“ (siehe Kapitel 3.3.2) mit Hilfe von verteilter Simulation umgesetzt werden könnte. Auch wird ein Algorithmus vorgestellt, der helfen soll optimale Fabrikstandorte zu finden (126).

Obwohl es sich um einen speziellen Anwendungsaspekt handelt, wurde vergleichsweise viel zu diesem Thema publiziert, was das Potenzial verdeutlicht. Der Unterschied liegt vielleicht darin, dass schon über viele Jahre hinweg in der Logistik, aber insbesondere in der Produktion, Simulationen genutzt werden, diese sich allerdings mehr auf das eigene Unternehmen beschränken. Bei virtuellen Fabriken (12, 36, 102) können schon im Voraus Planungen stattfinden, gegebenenfalls über Unternehmensgrenzen hinaus zu produzieren. Dies kann insbesondere vor dem Hintergrund des Datenschutzes und der Effizienz noch an Bedeutung gewinnen.

4.2.5 Militärische Anwendungen

Da die verteilte Simulation hier ihren Ursprung hat, gab es auch hier Publikationen zu potenziellen Anwendungsgebieten. Hier ist insbesondere Forschung betrieben worden, Gefechtssituationen realitätsnah darzustellen. Mit Hilfe von verteilter Simulation sollen möglichst komplexe und detailreiche Umgebungen simuliert werden, in denen sich viele Teilnehmer interaktiv begegnen sollen. Alle der vier wissenschaftlichen Arbeiten lassen sich grob so beschreiben. Ohne Zweifel hat verteilte Simulation in diesem Bereich Potenzial, allerdings spielen sie eine untergeordnete Rolle in den zugrunde liegenden Publikationen, sodass deutlich die Tendenz zu den zivilen Anwendungen geht. Die Beiträge zu diesem Thema sind auch nicht sehr innovativ, in dem Sinne, dass sie komplett neue Anwendungen präsentieren, denn sie fokussieren sich mehr auf den bekannten Themengebieten, wie die Realisierung von komplexen Gefechtsszenarien. Die Forschung der letzten Jahre scheint anderen Bereichen eine höhere Signifikanz zuzumessen.

4.2.6 Virtuelle Umgebungen

In dieser Kategorie kann man zwei Tendenzen erkennen, denn zum einen gibt es Beiträge, die komplexe virtuelle Welten erschaffen, damit Personen reale Szenarien, beispielsweise Gefechtssituation, trainieren können (92). Ergänzend gibt es noch Beiträge zu

Trainingsumgebungen in der Schifffahrt (131) oder Fahr simulatoren von PKW (132). Der zweite Blickwinkel, mit der die virtuellen Umgebungen betrachtet wird, ist wesentlich technischer, denn Themen wie die zeitliche Konsistenz (137), die Skalierbarkeit (127) oder Performance (108) von virtuellen Welten werden auch diskutiert.

Virtuelle Umgebungen spielen in wesentlich geringerem Maße eine Rolle als vielleicht zu Beginn der Forschung angenommen wurde. Seit 2005 konnten nur 3 Beiträge (92, 108, 127) diesem Themengebiet zugeordnet werden, sodass die Relevanz über die Jahre abgenommen hat. Virtuelle Welten werden oft im Kontext der verteilten Simulation erwähnt, allerdings seltener als andere Aspekte. Auch stehen viele Beiträge im Zusammenhang mit Gefechtssimulationen (92) oder es werden die oben erwähnten technischen Aspekte analysiert, sodass kaum neue Anwendungsbereiche der virtuellen Welten erschlossen wurden. Die Chancen virtuelle Welten mit Hilfe von verteilter Simulation zu realisieren ist durchaus vorhanden, allerdings nur für sehr spezifische Fälle wie Trainingsumgebungen.

4.2.7 Interdisziplinäre Beiträge

Aufgrund der Vielfalt der Publikationen ist diese Kategorie nur sehr schwer zusammenzufassen, aber auch hier lässt sich, ähnlich bei der Kategorie „Virtuelle Umgebungen“, in praxisbezogene Arbeiten und allgemein wissenschaftlichen Arbeiten eine Einteilung vornehmen. Insgesamt 5 Beiträge (62, 63, 84, 81, 118) haben einen direkten Anwendungsbezug zu globalen Krisen und beschäftigen sich mit der Frage, warum verteilte Simulation in diesem Zusammenhang eine adäquate Lösung bieten kann. Basierend auf Umfragen (68, 69, 83, 88) oder Befragungen (18) verschiedener Experten, wird das Zukunftspotenzial eingeschätzt. Ergänzend dazu wird eine sehr weite Bandbreite abgedeckt, wie Sicherheitsprobleme, also das unbefugte Eindringen in HLA Systeme („Access Control“, 11), neue Nutzerinterfaces (145), Ad-hoc verteilte Simulation (120, 106) oder spezielle (agentenbasierte) Simulationsprogramme (85), um nur einige Beispiele zu nennen.

Hier konkrete Prognosen abzugeben gestaltet sich als schwierig, da sich Beiträge und Themen sehr stark unterscheiden. Vielmehr ist zu erkennen, dass zumindest bei den Anwendungen weitere Szenarien für verteilte Simulation durchaus geeignet sind. Die Vielfalt anderer Beiträge (siehe Anhang) zeigt, dass es ein sehr breites Forschungsspektrum gibt, allerdings konnte kein spezielles Szenario identifiziert werden, welchem ein sehr großes Potenzial oder Entwicklungschancen zugewiesen werden können.

4.2.8 Interoperabilität

Aufgrund der Idee von verteilter Simulation, nämlich der Kombination von verschiedenen Teilsimulationen in einer Gesamtsimulation, spielt das Thema der Interoperabilität von Simulationsprogrammen eine zentrale Rolle. Im Speziellen hat die HLA eine gute Infrastruktur geschaffen, mit der dann schneller Teilsimulationen kommunizieren oder Objekte übergeben werden können. Es gibt allerdings 12 Beiträge, die sich damit befassen wie verschiedene Simulationsprogramme kommunizieren und interagieren. Beispiele sind 71, 27 oder 31. Dies zeigt, dass Interoperabilität von heterogenen Simulationssoftware sehr wichtig ist und dass es trotz HLA keine „Plug-and Play“ Synchronisation gibt. Auffallend ist, dass die Mehrheit der

Beiträge bis 2007 verfasst wurde, sodass man bis zu diesem Zeitpunkt die Interoperabilität als ein zentrales Forschungsgebiet ansehen kann. Dies wird insbesondere durch die beschriebenen Problemstellungen in der Logistik und der Produktion deutlich, sodass dieses Thema als äußerst wichtig beschrieben werden kann.

4.2.9 Performanceanalyse

Insgesamt 32 Beiträge befassen sich direkt oder indirekt damit die Performance von verteilter Simulation zu verbessern. In diese Kategorie fallen also Beiträge, die Ansätzen vorstellen, die die Rechengeschwindigkeit erhöhen können. Es werden Probleme wie der Lookaheadwert (29) oder das optimale Nutzen von vielen Prozessoren (16) diskutiert. Ergänzend dazu werden technische Aspekte wie der TIME WRAP (20, 30), die Skalierbarkeit (127) oder die Auswirkungen einer verbesserten RTI thematisiert (66). Jede technische Änderung wird hinsichtlich ihrer Leistungsfähigkeit getestet, sodass viele Publikationen zumindest indirekt dieser Kategorie zugeordnet werden können.

Die zahlreichen Forschungsaspekte verdeutlichen, dass es für verteilte Simulation in bestimmten Szenarien eine Herausforderung darstellt die gewünschte Performance, wegen der oben beschriebenen Probleme, zu erreichen. Alternativ wird auch gezeigt, warum gerade verteilte Simulation dazu geeignet ist, eine bessere Performance zu erreichen (Beispiel: 123).

4.2.10 Forschung zur grundlegenden IT-Architektur

Ähnlich wie bei den anderen Kategorien ist auch hier der Inhalt der Artikel sehr differenziert zu betrachten, da sie sich mit sehr unterschiedlichen Aspekten der grundlegenden Technik der verteilten Simulation befassen. Folgende Bereiche hat dabei eine starke Relevanz:

- MISSION Projekt (im Jahr 2000) (133,134,135, 5)
- Verteilte Simulation mittels GRIDS (3) und andere Alternativen (21)
- Neue oder verbesserte Middleware (58, 66, 41, 95, 100)
- Neue Algorithmen (99, 75)
- Verteilte Simulation mit Hilfe von Multiagentensystemen (7, 26, 98)

Wie man erkennen kann, befassen sich die Autoren dieser Beiträge mit sehr grundsätzlichen Themen wie HLA, denn diese hat sich weitgehend als Standard durchgesetzt, allerdings wird sie trotzdem nicht immer verwendet. Die HLA liefert die grundlegende Architektur, die beispielsweise die Kommunikation der Federates mit Hilfe des RTI gewährleistet, sodass das Nutzen diesen Standards viele Vorteile nach sich zieht. Allerdings bedarf die Middleware einigen Änderungen, denn viele Wissenschaftler haben Beiträge zu der Verbesserung der RTI publiziert. Beispiele wäre die Publikation 66, die die Middleware „Starlink“ beschreibt, welche eine weitaus größere Anzahl an Federates in einer verteilten Simulation einbinden kann, als die vorher genutzten RTI. Des Weiteren wird auch sehr oft erwähnt, wie Multiagentensysteme verteilte Simulation realisieren können. Die in der Aufzählung erwähnte „neuen Algorithmen“ sollen dabei helfen den Daten- und Informationsaustausch bei verteilter Simulation zu verbessern (107) oder dem Zeitkonsistenzproblem (99) zu begegnen. Im Jahr 2000 war das MISSION Projekt noch sehr stark repräsentiert, da die Forschung dazu Ende der 90er Jahre stattfand. Ähnlich wie die HLA kann die GRIDS Architektur für verteilte Simulation genutzt

werden, allerdings gibt es nur wenige Beiträge, die spezielle GRIDS der HLA vorziehen. Hier ist insbesondere Beitrag 3 zu nennen. Neu und interessant ist die Entwicklung der sogenannten „HLA-Cloud“, welche im Jahr 2013 in dem Beitrag „A SaaS-based Automated Framework to Build and Execute Distributed Simulations from SysML Models“ von Bocciarelli et al. (117) beschrieben wurde.

Weiter ist zu erkennen, dass in dieser Kategorie so gut wie gar keine Beiträge konkrete Anwendungen beschreiben. Vielmehr ist es so, dass die zugrunde liegende Architektur verbessert werden soll, sodass im Folgenden verteilte Simulation für reale Systeme genutzt werden kann. Fast keiner dieser Publikationen bezieht sich daher direkt auf die Logistik, Produktion oder eine Verkehrssimulation, welche Entwicklungen und Anwendungen ja durchaus in anderen Publikationen diskutiert werden.

4.3 Auswertung und Schlussfolgerungen

Die Untersuchung der Beiträge hat einige Erkenntnisse geliefert. Zum einen ist zu erkennen, dass verteilte Simulation über den gesamten Zeitraum (2000 bis zum Jahr 2015) hinweg ein Potenzial für zivile Anwendungen zugerechnet wird und es gibt auch einige konkrete Anwendungen. Schon im Jahr 2000 gab es einen Fahrsimulator (siehe Tietje et al., 2000), der mit Hilfe der HLA mit anderen Simulationen interagieren soll, sodass man grundsätzlich eine Entwicklungstendenz zu mehr speziellen und zivilen Anwendungen hat. In den betrachteten 15 Jahren stehen Anwendungen für die Logistik oder Produktion im Raum, allerdings überwiegt in jedem Jahr die Forschung der zugrunde liegenden Technik, wie die zur Middleware. Da auch Alternativen zu der HLA vorgeschlagen werden, hier sind insbesondere die Beiträge 19 und 21 zu nennen, scheint dieser Standard nicht generell akzeptiert zu sein, allerdings ist auch zu sagen, dass in der Mehrheit der Arbeiten HLA genutzt wird. Vorgeschlagene Alternativen wie D-Sol (21), welche eine andere Architektur darstellt, oder GRIDS, werden nicht über Jahre in den Fachtagungen erwähnt, sodass sie sich nicht gegenüber der HLA durchzusetzen scheinen. Die meisten Entwicklungen basieren noch auf der HLA, wobei durch die zahlreichen Beiträge zu einer verbesserten Middleware (41, 100, 95 und weitere) noch erhebliches Entwicklungspotenzial gesehen wird. Auch die Interoperabilität, insbesondere von Standardsimulationssoftwarepaketen ist ein zentrales Forschungsgebiet, insbesondere in den Jahre von 2000 bis zu Jahr 2007 wurden dazu 11 Beiträge verfasst, was die Relevanz dieses Themas hervorhebt. In den folgenden Jahren gab es dann keine Beiträge mehr, die sich explizit mit der Interoperabilität von COTS Simulationssoftware beschäftigen. Dies lässt die Deutung zu, dass bei der Interoperabilität von Softwarepaketen mit Hilfe der HLA Fortschritte erzielt wurden. Multiagentensysteme und deren Nutzen für verteilte Simulation ist auch ein sehr beachtetes Forschungsgebiet, denn insgesamt 25 Beiträge (7, 26, 35, 41, 42, 80, 81, 85, 91, 94, 124, 129, 146, 32, 98, 101, 111, 112, 118, 138, 143, 147, 38, 62, 64) befassen sich direkt oder indirekt mit Multiagentensystemen und deren Einfluss auf die verteilte Simulation. Beiträge zu Multiagentensystemen fallen auch im gesamten Zeitraum an, sodass kontinuierlich an diesem Ansatz geforscht wird. Diese Erkenntnisse wurden durch die inhaltlich Untersuchung durch Eingliederung der Beiträge der ASIM sowie der Winter Simulation Conference gewonnen:

- 1) Anwendungen spielen, insbesondere im Vergleich zu der Grundlagenforschung, eine eher untergeordnete Rolle, allerdings zeigen die konkreten Anwendungsbeispiele, die präsentiert werden, dass verteilte Simulation weit mehr als ein theoretischer Ansatz ist und für manche Problemstellungen ein adäquate Lösung darstellt.
- 2) Die HLA hat sich weitgehend als Standard durchgesetzt, wobei auch Publikationen andere Architekturen für verteilte Simulationen vorschlagen. Von denen hat sich allerdings keine gegenüber der HLA durchgesetzt, sodass man HLA als recht verbreiteten Standard ansehen kann.
- 3) Aufgrund der Quantität der wissenschaftlichen Beiträge, die sich mit einer verbesserten Middleware auseinandersetzen, kann diese als eine Schwachstelle der HLA angesehen werden.
- 4) Die Hauptanwendungsgebiete der verteilten Simulation werden für komplexe und interaktive Systeme, welche insbesondere in Logistik, aber auch in der Produktion, der Verkehrssteuerung oder der Fabrikplanung anzutreffen sind, gesehen. Virtuelle Welten, die für Trainingsszenarien genutzt werden, sind bedingt durch ihren Simulationsaufwand sehr gut geeignet, mittels verteilter Simulation eine Vielzahl von Teilnehmern interagieren zu lassen. Auch könnte verteilte Simulation helfen bei globalen Katastrophen, wie etwa Epidemien, geeignete Gegenmaßnahmen zu modellieren und anzuwenden.
- 5) Die Multiagentensysteme nehmen in dem gesamten Zeitraum eine zentrale Rolle ein und sind ein vielbeachtetes Forschungsgebiet.
- 6) Verteilte Simulation wird auch für militärische Zwecke erforscht, allerdings steht die Anwendung in industriellen und zivilen Bereichen klar im Vordergrund. Mit Ausnahme von wenigen Beiträgen wird der Fokus auf zivile Anwendungen gelegt.
- 7) Die Interoperabilität zwischen Standardsimulationssoftware hat bedingt durch die HLA Fortschritte erzielen können.
- 8) Die Performance von verteilter Simulation ist ein viel betrachtetes Problem, insbesondere die Zeitinkonsistenz der einzelnen Teilsimulationen ist problematisch. Auch das optimale Nutzen von Prozessorenkapazitäten stellt ein oft beschriebenes Forschungsgebiet dar.
- 9) Die Einschätzung, insbesondere am Ende der 90er Jahre, sind als zu optimistisch anzusehen, denn die Industrie sieht zwar vielfach die Vorteile der verteilten Simulation, allerdings ist die reale Anwendungen noch recht selten. Dies zeigt sich beispielsweise in einer Umfrage aus dem Jahr 2006 (79).

Die obigen Erkenntnisse und die zahlreichen Anwendungsbeispiele zeigen, dass verteilte Simulation Zukunftspotenzial besitzt. Ebenso wurde aber noch kein Reifegrad erreicht, der ermöglicht, dass verteilte Simulation für viele verschiedene Szenarien eingesetzt wird. Allerdings ist auch zu erwähnen, dass es ausreichend viele Beispiele gibt, die durch verteilte Simulation schneller simuliert werden oder neue Lösungsansätze generieren können.

Literaturverzeichnis

- Daniluk, D.; ten Hompel, M.: Verteilte Simulation und Emulation von Materialflusssystemen mit dezentraler Steuerung. In: Rabe, M. und Clausen, U. (Hg.): *Simulation in Production and Logistics*, 16. ASIM-Fachtagung Simulation und Logistik, Stuttgart. Fraunhofer Verlag, 2015, S. 259–267.
- Dobermann, M.; Stelbe, U.: Eine verteilte maritime Simulations-Trainingsumgebung auf Basis der HLA. In: Rabe, M. und Mertins, K. (Hg.): *The New Simulation in Production and Logistics. Prospects, Views and Attitudes*. 9. ASIM-Fachtagung Simulation in Produktion und Logistik, Berlin. Fraunhofer IPK, 2000, S. 347–356.
- Duinkerken, M.B.; Ottjes, J.A.; Lodewijks, G.: The Application of Distributed Simulation in THOMAS: Redesigning a Complex Transport Model. In: Yücesan, E.; Chen, C-H.; Snowdon, J.L. und Charnes, J.M. (Hg.): *Proceedings of the 2002 Winter Simulation Conference, San Diego, USA, 2002*, S. 1207–1213.
- Fujimoto, R.M.: *Parallel and Distribution Simulation Systems*. New York: Wiley-Interscience Verlag, 2000
- Gabler Wirtschaftslexikon, Springer Gabler Verlag, Stichwort: Reihenfolgeplanung, online im Internet: <http://wirtschaftslexikon.gabler.de/Archiv/72931/reihenfolgeplanung-v5.html>.
- Hintze A.; Schumann M.; Stüring, S.: Flexibles Modellieren von Maschinen und Anlagen in verteilten Umgebungen. In: Rabe, M. und Mertins, K. (Hg.): *The New Simulation in Production and Logistics. Prospects, Views and Attitudes*. 9. ASIM-Fachtagung Simulation in Produktion und Logistik, Berlin. Fraunhofer IPK, 2000, S. 339–347.
- Juan A.; Marull, J.M.; Jorba, J.; Hester, J.: Using Agent-based Simulation and Distributed Computing to solve Vehicle Routing Problems. In: Stock, P. und Zülch, G. (Hg.): *Integrationsaspekte der Simulation: Technik, Organisation und Personal*. 14. ASIM-Fachtagung Simulation in Produktion und Logistik, Karlsruhe. KIT Scientific Publishing, 2010, S. 523–530.
- Kalasky, D.; Coffman, M.; De Grano, M.; Field, K. (2010): Simulation-Based Manpower Planning with Optimized Schedule Distributed Multi-User Environment. In: Johansson, B.; Jain, S., Montoya-Torres, J.; J. Hagan, J. and Yücesan, E. (Hg.): *Proceedings of the 2010 Winter Simulation Conference, Baltimore, USA, 2010*, S.3447–3599.
- Klein, U., S. Straßburger: Zivile Anwendungspotentiale der High Level Architecture (HLA). *Symposium Neue Technologien in der wehrtechnischen Simulation, Mannheim, 1997*, Online verfügbar unter <http://strassburger-online.de/papers/Mannheim97.pdf>, zuletzt geprüft am 14.09.2015.
- Law, A.M.; Kelton, W.D.: *Simulation Modeling and Analysis*. New York, McGraw-Hill Verlag, 2000
- Lendermann, P.: About the Need for Distributed Simulation Technology for the Resolution of Real-World Manufacturing and Logistics Problems. In: Perrone, L.F.; Wieland, F.P.; Liu, L.; Lawson, B.G.; Nicol, D.M. und Fujimoto, R.M. (Hg.): *Proceedings of the 2006 Winter Simulation Conference, Monterey, USA, 2006*, S.1119-1128
- Logan B.; Theodoropoulos G.: The Distributed Simulation of Multi-Agent Systems. *Nottingham University*. 2001, Online verfügbar unter <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.1221&rep=rep1&type=pdf>, zuletzt geprüft am 28.09.2015.

- Özgün, O.; Barlas, Y.: Discrete vs. Continuous Simulation: When Does It Matter? In: *Proceedings of the 27th. International Conference of The System Dynamics Society. Albuquerque, NM, USA, 2009*, Online verfügbar unter <http://www.systemdynamics.org/conferences/2009/proceed/papers/P1199.pdf>, zuletzt geprüft am 19.09.2015.
- Parashar, M.; Lee, C.A.: GRID Computing: Introduction and Overview. *IEEE. 2005*, Online verfügbar unter <http://coewww.rutgers.edu/www4/cacweb/TASSL/Papers/proc-ieee-intro-04.pdf>, zuletzt geprüft am 03.10.2015.
- Raab, M.; Schulze, T.; Straßburger, S.: Management of HLA-Based Distributed Legacy SLX-Modell. In: Mason, S.J.; Hill, R.R.; Moench, L. und Rose, O. (Hg.): *Proceedings of the 2008 Winter Simulation Conference, Miami, USA, 2008a*, S. 1086–1093.
- Raab, M.; Schulze, T.; Straßburger, S.: Erfahrungen aus der Anwendung von HLA-basierter verteilter Simulation im Nutzfahrzeugbereich In: Rabe, M. (Hg.): *Advances in Simulation for Production and Logistics Applications, 13. ASIM-Fachtagung Simulation in Produktion und Logistik, Stuttgart. Fraunhofer IRB-Verlag, 2008b*, S. 499–508.
- Rabe, M.: Future of Simulation in Production and Logistics: Facts and Visions. Simulation wohin? Fakten und Visionen. In: Rabe, M. und Mertins, K. (Hg.): *The New Simulation in Production and Logistics. Prospects, Views and Attitudes. 9. ASIM-Fachtagung Simulation in Produktion und Logistik, Berlin. Fraunhofer IPK, 2000*, S. 21–42.
- Robinson, S.: Simulation. The Practice of Model Development and Use. *Chichester. John Wiley & Sons Ltd. Verlag, 2004*
- Schulze, T.; Lantzsch, G; Klein, U.; Straßburger, S.: Interoperabilität zwischen Simulationsmodellen auf Basis der High Level Architecture. In: Mertins, K. und Rabe, M. (Hg.). *Erfahrungen aus der Zukunft. 8. ASIM-Fachtagung Simulation in Produktion und Logistik, IPK Berlin 1998*, S. 369-379.
- Schumann, M.: Zeit- und Konsistenzmanagement verteilter, interaktiver Echtzeitanwendung am Beispiel einer virtuellen Trainingsumgebung. In: M. Rabe, M. und Mertins, K. (Hg.): *Experiences from the Future. New Methods and Applications in Simulation for Production and Logistics, 11. ASIM-Fachtagung Simulation in Produktion und Logistik, Stuttgart. Fraunhofer IRB-Verlag, 2004*, S. 423–432
- Siebers, P.; Aickelin, U.: Introduction to Multi-Modal Agent Simulation, *University of Nottingham, 2008*. Online verfügbar unter <http://arxiv.org/ftp/arxiv/papers/0803/0803.3905.pdf>, zuletzt geprüft am 28.09.2015.
- Soliman, H.M.: Frameworks for Component-based Simulation. *Department of Information Systems, College of Computer & Information Sciences. Riyadh, Saudi Arabia. 2000*, Online verfügbar unter http://ac.els-cdn.com/S1319157801800045/1-s2.0-S1319157801800045-main.pdf?_tid=dbd45974-65da-11e5-bbb2-00000aab0f26&acdnat=1443442817_b0eab39aefb41f110df367b0c47410e4, zuletzt geprüft am 28.09.2015.
- Straßburger, S.; Schulze, T.: Zukunftstrends in den Bereichen Verteilte Simulation und Verteilte Virtuelle Umgebungen. In: Rabe, M. (Hg.): *Advances in Simulation for Production and Logistics Applications, 13. ASIM-Fachtagung Simulation in Produktion und Logistik, Stuttgart. Fraunhofer IRB-Verlag, 2008*, S. 489–498.
- Suzumura, T.; Kanezashi, H.: Multi-Modal Traffic Simulation Platform on Parallel and Distributed Systems. In: Buckley, S.J. und Miller, J.A. (Hg.): *Proceedings of the 2014 Winter Simulation Conference. Savannah, USA, 2014*, S. 769–780.
- Taylor, S.; Tan, G.; Zhao, N.: Automobile Manufacturing Supply Chain Simulation in the GRIDS Environment. In: Chick, S.; Sánchez, P.J.; Ferrin, D. und Morrice, D.J. (Hg.): *Proceedings of the 2003 Winter Simulation Conference, 2003. New Orleans, USA, 2003*, S. 1149–115

Tietje, H.; Klein, U.; Straßburger, S.: Demonstration von HLA-basierten verteilten Simulationsmodellen. In: Rabe, M. und Mertins, K. (Hg.): *The New Simulation in Production and Logistics. Prospects, Views and Attitudes*. 9. ASIM-Fachtagung Simulation in Produktion und Logistik, Berlin. Fraunhofer IPK, 2000, S. 357–364.

Venkateswaran, J.; Jafferli, M.Y.K.; Son, Y.-J.: Distributed Simulation: An Enabling Technology for the Evaluation of Virtual Enterprises. In: Peters, B.A.; Smith, J.S.; Medeiros, D.J. und Rohrer, M.W. (Hg.): *Proceedings of the 2001 Winter Simulation Conference*. Arlington, USA, 2001, S.856-862

Wild, R.: *Essentials of Operations Management*. 5. Auflage, London, 2002, Continuum Verlag.

Abbildungsverzeichnis

<i>Abbildung 1: Drei-Phasen Methode, Robinson 2004.....</i>	<i>5</i>
<i>Abbildung 2: Hardware zur Realisierung verteilter und paralleler Simulation, Fujimoto 2000.....</i>	<i>8</i>
<i>Abbildung 3: Technik der High Level Architecture, Fujimoto 2000.....</i>	<i>9</i>
<i>Abbildung 4: Supply Chain Netzwerk, Lendermann 2006.....</i>	<i>22</i>
<i>Abbildung 5: Hauptmodelle der Hafensimulation, Duinkerken 2002.....</i>	<i>24</i>
<i>Abbildung 6: "Borderless Fab", Lendermann 2006.....</i>	<i>26</i>
<i>Abbildung 7: Materialfluss in einer Automobilfabrik, Raab et al.2008a.....</i>	<i>26</i>
<i>Abbildung 8: HLA-basierte Interaktion von Simulationsmodellen, Raab et al. 2008a.....</i>	<i>27</i>
<i>Abbildung 9: Komponenten der Verkehrssimulation, Suzumura und Kanezashi 2014.....</i>	<i>28</i>
<i>Abbildung 10: Anwendungspotenzial verteilter Simulation, Straßburger und Schulze 2008.....</i>	<i>30</i>
<i>Abbildung 11: Verbreitungsgrad verteilter Simulation, Straßburger Schulze 2008.....</i>	<i>31</i>
<i>Abbildung 12: Themenübersicht der analysierten wissenschaftlichen Beiträge.....</i>	<i>33</i>
<i>Abbildung 13: Quantitative Zuordnung der Publikationen in ihre thematischen Kategorien.....</i>	<i>34</i>

Tabellenverzeichnis

<i>Tabelle 1: HLA Regeln, Fujimoto 2000.....</i>	<i>10</i>
<i>Tabelle 2: ankommende Flüge in Flughafensimulation.....</i>	<i>17</i>
<i>Tabelle 3: Anzahl der wissenschaftlichen Publikationen zum Thema verteilter Simulation.....</i>	<i>32</i>

Abkürzungsverzeichnis

ABS	Agent-Based Simulation
CORBA	Common Object Request Broker Architecture
COTS	Commercial of the Shelf, (Standardsimulationssoftware)
GRIDS	Generic Runtime Infrastructure for Distributed Simulation
HLA	High Level Architecture
IDL	Interface Definition Language
LP	Logischer Prozess
OEMT	Object Exchange Model Template
ORB	Object Request Broker
RTI	Run-Time-Interface
VE	Virtual Environments
VIS	Visual Interactive Simulation

Abkürzungen für den Anhang sowie Abbildung 13

ASIM	Fachtagungen der ASIM (Arbeitsgruppe Simulation)
FP	Fabrik- und Serviceplanung
INT	Interoperabilität
IT	Forschung zur grundlegenden IT-Infrastruktur
LOG	Logistische Anwendungen
PER	Performanceanalyse
PRO	Anwendung in der Produktion
Mili	Militärische Anwendungen
VE	Virtuelle Umgebung
VS	Verkehrssteuerung
W	Weitere interdisziplinäre Beiträge
WIN	Winter Simulation Conference

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
1	Creating Distributed Simulation Using DEVS M&S Environments	2000	WIN	Bernard P. Zeigler and Hessam S. Sarjoughian	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Analysis of the Virtual Enterprise Using Distributed Supply Chain Modeling and Simulation: An Application of e-SCOR	2000	WIN	Michael W. Barnett and Charles J. Miller	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Distributed Supply Chain Simulation in GRIDS	2000	WIN	Rajeev Sudra, Simon J. E. Taylor, and Tharumasegaram Janahan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	Distributed Supply Chain Simulation Across Enterprise Boundaries	2000	WIN	Boon Ping Gan, Li Liu, Sanjay Jain Stephen J. Turner, Wentong Cai, and Wen-Jing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	The IMS Mission Architecture for Distributed Manufacturing Simulation	2000	WIN	Charles McLean and Frank Riddick	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
6	Neutral Template Libraries for Efficient Distributed Simulation within a Manufacturing System Engineering Platform	2000	WIN	Kai Mertins, Markus Rabe, and Frank-Walter Jaekel	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
7	An Agent-Based Framework for Linking Distributed Simulations	2000	WIN	Linda F. Wilson, Daniel Burroughs, Jeanne Sucharitaves, and Anush	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8	Parallel and Distributed Simulation Systems	2001	WIN	Richard M. Fujimoto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Distributed Simulation and Control: The Foundations	2001	WIN	Wayne J. Davis	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
10	The Economic Effects of Reusability on Distributed Simulations	2001	WIN	Mary Ewing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	Security Issues in High Level Architecture Based Distributed Simulation	2001	WIN	Asa Elkins, Jeffery W. Wilson , Denis Gracanin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	Distributed Simulation: An Enabling Technology for the Evaluation of Virtual Enterprises	2001	WIN	Jayendran Venkateswaran, Mohammed Yaseen Kalachikan Jafferli, and	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13	Distributed Simulation with Incorporated APS Procedures for High-Fidelity Supply Chain Optimization	2001	WIN	Peter Lendermann and Boon Ping Gan and Leon F. McGinnis	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14	Towards COTS Distributed Simulation Using GRIDS	2001	WIN	Simon J.E. Taylor, Rajeev Sudra, and Tharumasegaram Janahan , Gary Tan and John	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	On the Response Surface Methodology and Designed Experiments for Computationally Intensive Distributed Aerospace Simulations	2002	WIN	Paul Stewart, Peter J. Fleming and Sheena A. MacKenzie	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16	Load Sharing in Heterogeneous Distributed Systems	2002	WIN	Helen D. Karatza and Ralph C. Hilzer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	UML based Modeling of Performance Oriented Parallel and Distributed Applications	2002	WIN	Sabri Pillana and Thomas Fahringer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
18	Distributed Simulation and Industry: Potentials and Pitfalls	2002	WIN	Simon J. E. TaylorAgostino Bruzzone Richard Fujimoto, Boon Ping Gan Steffen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
19	Distributed Spatio-Temporal Modeling and Simulation	2002	WIN	Thomas Schulze , Andreas Wytzisk and Ingo Simonis and Ulrich Raape	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	Managing External Workload with BSP Time Warp	2002	WIN	Malcolm Yoke Hean Low	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	D-SOL; A Distributed Java based Discrete Event Simulation Architecture	2002	WIN	Peter H.M. Jacobs, Niels A. Lang, and Alexander Verbraeck	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
22	Manufacturing Adapter of Distributed Simulation Systems Using HLA	2002	WIN	Hironori Hibino , Yoshiro Fukuda , Yoshiyuki Yura , Keiji Mitsuyuki and Kiyoshi Kaneda	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>					
23	The Application of Distributed Simulation in TOMAS: Redesigning a Complex Transportation Model	2002	WIN	Mark B. Duinkerken, Jaap A. Ottjes, and Gabriel Lodewijks	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24	Development of Distributed Simulation Model for the Transporter Entity in a Supply Chain Process	2002	WIN	Richard J. Linn, Chin-Sheng Chen, and Jorge A. Lozan	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25	Distributed Simulation Systems	2003	WIN	Richard M. Fujimoto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26	SPADES – A Distributed Agent Simulation Environment with Software-in-the-Loop Execution	2003	WIN	Patrick F. Riley and George F. Riley	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
27	Distributed Simulation with COTS Simulation Packages	2003	WIN	Csaba Attila Boer and Alexander Verbraeck	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
28	Time Management Issues in COTS Distributed Simulation: A Case Study	2003	WIN	Simon J.E. Taylor , Jon Sharpe and John Ladbrook	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
29	Synchronization and Management of Shared State in HLA-Based Distributed Simulation	2003	WIN	Boon Ping Gan and Malcolm Yoke Hean Low and Junhu Wei, Xiaoguang Wang,	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
30	Causal Order based Time Warp: A Tradeoff of Optimism	2003	WIN	Yi Zeng, Wentong Cai, and Stephen J. Turner	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
31	HLA-CSPiF Panel on Commercial Off-the-Shelf Distributed Simulation	2003	WIN	Simon J.E. Taylor , Boon Ping Gan , Steffen Straßburger and Alexander Verbraeck	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
32	UTSAF: A Multi-Agent-Based Framework for Supporting Military-Based Distributed Interactive Simulations in 3D Virtual Environments	2003	WIN	Joseph Manojlovich, Phongsak Prasithsangaree, Stephen Hughes, Jinlin Chen, and	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
33	Pre-Reckoning Algorithm for Distributed Virtual Environments	2003	WIN	Thomas P. Duncan and Denis Gracanin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
34	Automobile Manufacturing Supply Chain Simulation in the GRIDS Environment	2003	WIN	Gary Tan and Na Zhao and Simon J.E. Taylor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
35	EPOCHS: Integrated Commercial Off-the-Shelf Software for Agent-Based Electric Power and Communication Simulation	2003	WIN	Kenneth M. Hopkinson and Kenneth P. Birman , Renan Giovanini and Denis V. Coury	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
36	Simulation of Distributed Manufacturing Enterprises: A New Approach	2003	WIN	Sameh M. Saad, Terrence Perera, and Ruwan Wickramarachchi	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
37	Distributed Supply Chain Simulation Using a Generic Job Running Framework	2003	WIN	Haifeng Xi, Heng Cao, Leonard Berman, and David Jensen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
38	A Framework for Adaptive Synchronization of Distributed Simulations	2004	WIN	Bertan Altuntas and Richard Allen Wysk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
39	An Automatic Distributed Simulation Environment	2004	WIN	Sarita Mazzini Bruschi, Regina Helena Carlucci Santana, Marcos José Santana, and	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
40	Incremental HLA-Based Distributed Simulation Cloning	2004	WIN	Dan Chen, Stephen John Turner, and Wentong Cai and Boon Ping Gan and Malcolm	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
41	Resolving Mutually Exclusive Interactions in Agent Based Distributed Simulations	2004	WIN	Lihua Wang, Stephen John Turner, and Fang Wang	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
42	Data Dissemination Techniques for Distributed Simulation Environments	2004	WIN	Bryan Horling and Victor Lesser	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
43	Analysis and Enhancement of Planning and Scheduling Applications in a Distributed Simulation Testbed	2004	WIN	Nirupam Julka, Peter Lendermann, Chin Soon Chong, and Long-Foong Mike	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
44	Analysis of a Borderless Fab Scenario in a Distributed Simulation Testbed	2004	WIN	Peter Lendermann, Boon Ping Gan, and Yoon Loong LohHiap Keong Tan and Sip Khean	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
45	Analysis of a Customer Demand Driven Semiconductor Supply Chain in a Distributed Simulation Test Bed	2004	WIN	Chin Soon Chong, Peter Lendermann, and Boon Ping Gan and Brett Marc Duarte,	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
46	An Architecture for Distributed Simulation Games	2004	WIN	Stijn-Pieter A. van Houten and Peter H.M. Jacobs	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
47	A Distributed Computing Architecture for Simulation and Optimization	2005	WIN	Yijia Xu and Suvrajeet Sen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
48	Bias in Parallel and Distributed Simulation Systems	2005	WIN	Tobias Kiesling and Rachid El Abdouni Khayari and Johannes Lüthi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
49	Interoperating Autosched AP Using the High Level Architecture	2005	WIN	Boon Ping Gan, Peter Lendermann, and Malcolm Yoke Hean Low , Stephen J.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
50	Using Parallel and Distributed Computing to Increase the Capability of Selection Procedures	2005	WIN	E. Jack Chen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
51	Enabling 1,000,000-Entity Simulations on Distributed Linux Clusters	2005	WIN	Gene Wagenbreth ,Ke-Thia Yao,Dan M. Davis ,Robert F. Lucas homas D. Gottschalk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
52	A Framework for Fault-Tolerance in HLA-Based Distributed Simulations	2005	WIN	Martin Eklöf, Farshad Moradi and Rassul Ayani	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
53	Language Based Simulation, Flexibility, and Development Speed in the Joint Integrated Mission Model	2005	WIN	David W. Mutschler	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
54	Usability Study of the Virtual Test Bed and Distributed Simulation	2005	WIN	Jeffrey W. Dawson, Ping Cheng, and Yanshen Zhu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
55	Supporting the Vision for Space with Discrete Event Simulation	2005	WIN	Grant R. Cates and Mansooreh Mollaghasemi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
56	Simulation of Grid Computing Infrastructure: Challenges and Solutions	2005	WIN	Sugato Bagchi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
57	Time Management in Distributed Factory Simulation, a Case Study Using HLA	2005	WIN	Ke Wang, Sheng Xu, and Leon F. McGinnis	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
58	Data Consistency in a Large-Scale Runtime Infrastructure	2005	WIN	Buquan Liu, Huaimin Wang, and Yiping Yao	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
59	Addressing Complexity Using Distributed Simulation: A Case Study in Spaceport Modeling	2005	WIN	aebok Park, Reinaldo Moraga, and Luis Rabelo, Jeffrey W. Dawson and Mario Marin and	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
60	Parallel and Distributed Simulation: Traditional Techniques and Recent Advances	2006	WIN	Kalyan Perumalla	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
61	CIMS: A Framework for Infrastructure Interdependency Modeling and Analysis	2006	WIN	Donald D. Dudenhoeffer and May R. Permann and Milos Manic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
62	Distributed Simulation for Homeland SecurityAn HLA-Based Multiagent System for Optimized Resource Allocation After Strong Earthquakes	2006	WIN	Frank Fiedrich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
63	A Concept Prototype for Integrated Gaming and Simulation for Incident Management	2006	WIN	Sanjay Jain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
64	SASSY: A Design for a Scalable Agent-Based Simulation System Using a Distributed Discrete Event Infrastructure	2006	WIN	Maria Hybinette, Eileen Kraemer, Yin Xiong, Glenn Matthews, and Jaim Ahmed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
65	A Distributed Simulation Approach for Modeling and Analyzing Systems of Systems	2006	WIN	Abeer Tarief Sharawi, Serge N. Sala-Diakanda, Adam Dalton, Sergio Quijada, Nabeel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
66	Development of a Runtime Infrastructure for Large-Scale Distributed Simulations	2006	WIN	Buquan Liu, Yiping Yao, Jing Tao, and Huaiming Wang	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
67	Implementation of Time Management in a Runtime Infrastructure	2006	WIN	Buquan Liu, Yiping Yao, Jing Tao, and Huaimin Wang	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
68	Distributed Simulation in Industry – A Survey, Part 1 – the COTS Vendors	2006	WIN	Csaba Attila Boer , Arie de Bruin and Alexander Verbraeck	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
69	Distributed Simulation in Industry – a Survey, Part 2 – Experts on Distributed Simulation	2006	WIN	Csaba Attila Boer , Arie de Bruin and Alexander Verbraeck	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
70	Optimistic-conservative Synchronization in Distributed Factory Simulation	2006	WIN	Leon McGinnis and Sheng Xu	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
71	Developing Interoperability Standards for Distributed Simulation and COTS Simulation Packages with the CSPI PDG	2006	WIN	Simon J. E. Taylor, Stephen J Turner, Malcolm Yoke Hean Low, and Xiaoguang Wang ,	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
72	About the Need for Distributed Simulation Technology for the Resolution of Real-world Manufacturing and Logistics Problems	2006	WIN	Peter Lendermann	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
73	Distributed Simulation with COTS Simulation Packages: A Case Study in Health Care Supply Chain Simulation	2006	WIN	Navonil Mustafee and Simon J. E. Taylor, and Korina Katsaliaki and Sally Brailsford	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
74	Symbiotic Simulation for Business Process Re-engineering in High-tech Manufacturing and Service Networks	2007	WIN	Malcolm Yoke Hean Low, Stephen John Turner, Ling Ding, and Hai Liang Peng Peter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
75	An Efficient Algorithm in the HLA Time Management	2007	WIN	Buquan Liu, Yiping Yao, and Huaimin Wang	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
76	Applying CSPI Reference Models for Factory Planning	2007	WIN	Steffen Strassburger, Thomas Schulze, Marco Lemessi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
77	Distributed Simulation for Interoperability Testing Along the Supply Chain	2007	WIN	Sanjay Jain, Frank Riddick, Andreas Craens	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
78	Panel: Distributed Simulation in Industry - a Real-world Necessity or Ivory Tower Fancy?	2007	WIN	Peter Lendermann, Leon F. McGinnis, Charles R. McLean, Simon	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
79	Applying Parallel and Distributed Simulation to Remote Network Emulation	2007	WIN	Yan Gu and Richard Fujimoto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
80A	A Flexible, Large-scale, Distributed Agent Based Epidemic Model	2007	WIN	Jon Parker	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
81A	A Teragrid-enabled Distributed Discrete Event Agent-based Epidemiological Simulation	2007	WIN	Douglas Roberts and Diglio A. Simoni	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
82	Hierarchical Distributed Simulation for 300mm Wafer Fab	2007	WIN	Sheng Xu and Leon F. McGinnis	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
83	Future Trends in Distributed Simulation and Distributed Virtual Environments: Results of a Peer Study	2008	WIN	Steffen Strassburger, Thomas Schulze ,Richard Fujimoto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
84	An Application of Parallel Monte Carlo Modeling for Real-Time Disease Surveillance	2008	WIN	David W Bauer Jr	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
85	Quantitative Assessment of an Agent-Based Simulation on a Time Warp Executive	2008	WIN	George Vulov, Tianhao He, and Maria Hybinette	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
86	Supporting Simulation in Industry Through the Application of Grid Computing	2008	WIN	Navonil Mustafee and Simon J. E. Taylor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
87	Management of HLA-Based Distributed Legacy SLX-Models	2008	WIN	Thomas Schulze , Steffen Strassburger and Michael Raab	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
88	Distributed Simulation in Industry – A Survey, Part 3 – The HLA Standard in Industry	2008	WIN	Csaba A. Boer and Arie de Bruin and Alexander Verbraeck	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
89	Service-Oriented-Architecture Based Framework for Multi-User Virtual Environments	2008	WIN	Xiaoyu Zhang and Denis Gracanin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
90	Aggregated 3D-Visualization of a Distributed Simulation Experiment of a Queuing System	2008	WIN	Wilhelm Dangelmaier, Matthias Fischer, Daniel Huber, Christoph Laroque,	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
91	Distributed Agent-Based Simulation of Construction Projects with HLA	2008	WIN	Hosein Taghaddos, Simaan AbouRizk, Yasser Mohamed, and Ivan Ourdev	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
92	Principles of Combat Modeling and Distributed Simulation	2009	WIN	Andreas Tolk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
93	Applying Web Services Technology to Implement Distributed Simulation for Supply Chain Modeling and Analysis	2009	WIN	Taejong Yoo, Kyungdoc Kim, Sunyoung Song, and Hyunbo Cho and Enver Yücesan	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
94	Scalability in Distributed Simulations of Agent-based Models	2009	WIN	Dirk Pawlaszczyk and Steffen Strassburger	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
95	Performing Distributed Simulation with Restful Web Services	2009	WIN	Khaldoon Al-Zoubi and Gabriel Wainer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
96	Services Modeling and Distributed Simulation DEVS / HLA Supported	2009	WIN	Gregory Zacharewicz, Thecle Alix, and Bruno Vallespir	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
97	Improving Modeling and Simulation Through Advanced Computing Techniques: Grid Computing and Distributed Simulation	2010	WIN	Simon Taylor , Navonil Mustafee , Shane Kite ,Steffen Strassburger , Stephen Turner	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
98	Interfacing Multi-Agent Models to Distributed Simulation Platforms: The Case of PDES-MAS	2010	WIN	Bart Craenen and Georgios K. Theodoropoulos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
99	A Fast Parallel Matching Algorithm for Continuous Interest Management	2010	WIN	Elvis S. Liu and Georgios K. Theodoropoulos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
100	Systems Engineering for Distributed, Live, Virtual, and Constructive (LVC) Simulation	2010	WIN	Scott Gallant (Effective Applications Corporation) and Chris Gaughan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
101	Maintaining a Distributed Symbiotic Relationship Using Delegate MultiAgent Systems	2010	WIN	Rutger Claes and Tom Holvoet	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
102	Developing Complex Distributed Simulation for Industrial Plant Construction using High Level Architecture Developing Complex Distributed Simulation for Industrial	2010	WIN	imaan AbouRizk, Yasser Mohamed, Hosein Taghaddos, Farzaneh Saba and Stephen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
103	3D CAD Modeling and Visualization of the Tunnel Construction Process in a Distributed Simulation Environment	2010	WIN	Yang Zhang, Elmira Moghani and Simaan AbouRizk and Siri Fernando	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
104	Simulation-Based Manpower Planning With Optimized Scheduling In A Distributed Multi-User Enviroment	2010	WIN	David Kalasky , Michael Coffmann , Melanie DeGrano and Kevin Field	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
105	Distributed Computing and M&S: Speeding up Simulation and Creating Large Models	2011	WIN	Simon Taylor, Steffen Strassburger	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
106	Statistical Issues in Ad Hoc Distributed Simulations	2011	WIN	Ya-Lin Huang, Wonho Suh, Christos Alexopoulos, Richard Fujimoto and Michael Hunter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
107	A Binary Partition-based Matching Algorithm for Data Distribution Management in the HLA/RTI	2011	WIN	Junghyun Ahn, Changho Sung and Tag Gon Kim	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
108	A Methodology for Managing Distributed Virtual Environment Scalabilit	2011	WIN	Lally Singh and Denis Gracanin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
109	Traces Generation to Simulate Large-Scale Distributed Applications	2011	WIN	Emilio Mancini and Olivier Dalle	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
110	Modelling and Simulation-based Design of a Distributed DEVS Simulator	2011	WIN	Eugene Syriani and Hans Vangheluwe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
111	On-the-fly Parallelization of Sequential Agent-Based Simulation Systems	2011	WIN	Cole Sherer , George Vulov and Maria Hybinette	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
112	Grid-based partitioning for large-scale distributed agent-based crowd simulation	2012	WIN	Yongwei Wang, Michael Lees and Wentong Cai	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
113	A Model-Driven Method For Building Distributed Simulation Systems from Business Process Models	2012	WIN	Paolo Bocciarelli, Daniele Gianni and Alessandra Pieroni and Andrea D'Ambrogio	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
114	FAST CONVERGING, AUTOMATED EXPERIMENT RUNS FOR MATERIAL FLOW SIMULATIONS USING DISTRIBUTED COMPUTING AND COMBINED METAHEURISTICS	2012	WIN	Christoph Laroque and Alexander Klaas, Jan-Hendrik Fischer and Mathis Kuntze	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
115	Model-driven Performance Prediction of HLA-Based Distributed Simulation Systems	2012	WIN	Daniele Gianni, and Paolo Bocciarelli and Andrea D'Ambrogio	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
116	Database-Driven Distributed 3D Simulation	2012	WIN	Martin Hoppen, Michael Schluse and Juergen Rossmann and Bjoern Weitzig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
117	A SaaS-based Automated Framework to Build and Execute Distributed Simulations from SysML Models	2013	WIN	Paolo Bocciarelli, Andrea D'Ambrogio and Andrea Giglio and Daniele Gianni	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
118	Distributed Hybrid Agent-Based Discrete Event Emergency Medical Services Simulation	2013	WIN	Anastasia Anagnostou, Athar Nouman and Simon J.E. Taylor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
119	Runtime Execution Management Of Distributed Simulations	2013	WIN	Chris Gaughan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
120	On the Transient Response of Open Queueing Networks Using Ad Hoc Distributed Simulations	2013	WIN	Ya-Lin Huang, Christos Alexopoulos, Michael Hunter and Richard Fujimoto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
121	Development of a Distributed Construction Project Management Game with COTS in the Loop	2013	WIN	Yasser Mohamed and Mostafa Ali	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
122	A Distributed Simulation Approach for Continuous Variable Control of a Home Care Crew Schedule Problem	2013	WIN	Seok Gi Lee (Pennsylvania State University)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
123	Multi-Modal Traffic Simulation Platform on Parallel and Distributed Systems	2014	WIN	Toyotaro Suzumura and Hiroki Kanezashi	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
124	Simulating Water, Individuals and Management Using a Coupled and Distributed Approach	2014	WIN	Jonathan Ozik, John T. Murphy and Nicholson T. Collier , Mark Altaweel ,	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
125	A DDS-Based Distributed Simulation Approach for Engineering-Level Models	2014	WIN	Dohyung Kim, Ockhyun Paek, Taeho Lee and Samjoon Park and Hyunshik Bae	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
126	Combining Biased Random Sampling with Metaheuristics for the Facility Location Problem in Distributed Computer Systems	2014	WIN	Guillem Cabrera, Sergio Gonzalez-Martin and Angel Alejandro Juan (Universitat	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
127	Controlling Scalability of Distributed Virtual Environment Systems	2014	WIN	Lally Singh , Denis Gracanin and Kresimir Matkovic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
128	Global Dynamic Load-Balancing for Decentralised Distributed Simulation	2014	WIN	Quentin Bragard, Anthony Ventresque and Liam Murphy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
129	Synchronization Methods for Distributed Agent Based Models	2014	WIN	Christine Harvey and James E. Gentile	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
130	Flexible Modelling of Machine Tools and Production Systems within Distributed Environments	2000	ASIM	Hintze, Schumann, Stüring	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
131	A distributed maritime simulation training environment	2000	ASIM	Dobermann, Stelbe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
132	Demostration of Simulation Models based on HLA	2000	ASIM	Straßburger, Klein	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
133	Architecture for Modeling and Simulation of Global Distributed Enterprises	2000	ASIM	Riddick, McLean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
134	Development of a Demonstrator for Modelling and Simulation of Global Distributed Enterprises	2000	ASIM	Jäkel, Arroyo Pinedo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
135	Neutral Template Libraries for Use in Globally Distributed Enterprises	2000	ASIM	Rabe, Friedland	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
136	The High level Architecture: Progress and Prospects	2000	ASIM	Dahlmann, Weatherly, Kuhl, Zabek	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
137	Managing Time and Consistency in distributed, Interactive real-time Applications using the example of virtual Traiing environmenz	2004	ASIM	Schumann	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
138	Coupling Architecture for a NET based MAS and a discrete Event based Simulator	2004	ASIM	Zimmermann, Mönhc	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
139	Interaction-Mechanism for a Cooperative Simulation of Material Flow Systems	2004	ASIM	Dangelmeier, Laroque, Renner, Mueck	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
140	Future Trends in Distributed Simulation and Distributed Virtual Environments Distributed Virtual Environments	2008	ASIM	Straßburger , Schulze	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
141	Experiences from Applying HLA-based Distributed Simulation in the Utility Vehicle Sector	2008	ASIM	Raab, Schulze, Straßburger	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
142	Verteilen von nebenläufigen Simulationsläufen über eine Service-orientierte Netzwerkstruktur	2008	ASIM	Dobler, Martin; Saler, Martin; März, Lothar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
143	Modelling of Multi-agent-based Material Flow Systems for Distributed Simulation	2010	ASIM	Daniluk, Damian; ten Hompel , Michael	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
144	A Distributed Virtual Environment for Qualification and Testing Work on the Logistics Systems in the Packaging Center	2010	ASIM	Zahn, Nico, Reitzer, Heiko	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

NR	Artikelname	Jahr	Org	Autoren	LOG	Pro	VS	FP	Mili	VE	W	INT	PER	IT
145	Supportsystem for Distributed Simulation	2010	ASIM	Lemessi, Marco Rehn, Gordon, Schulze, Thomas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
146	Using Agent-based Simulation and Distributed Computing to solve Vehicle Routing Problems	2010	ASIM	Angel A. Juan, Josep M. Marull, Josep Jorba, Josh Hester,	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
147	Distributed Simulation and Emulation of Material Flow Systems with Decentralized Control	2015	ASIM	ten Hompel, Daniluk	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					