Zukunftstechnologien in der Modellierung und Simulation

- **Java/Web**
  - Das Java-Phänomen
  - Java-basierte Simulation
    - Online-Simulation and -Animation
    - Java-basierte Simulatoren
    - Verteilte Simulation
- **Komponenten**
  - High Level Architecture (HLA)
    - Überblick
    - Beispiel

Using Java for Simulation

Outline

- Introduction
- Overview of Java
- Java-based simulation
  - Online simulation and animation
  - Simulation support packages
  - Distributed simulation
- Examples and experiences
- Conclusions

The Java Phenomenon

Java has emerged as a major force in the computing landscape

- as a programming language for the 90s
- as a new platform for the development of heterogeneous network-centric systems

in an astonishingly short period of time.

- Since May 1995 millions of copies of the JDK have been downloaded over the Internet
- Today almost every computer and telecommunication company has licensed Java technology from Sun Microsystems
- 14 000 people attended the 1998 JavaOne Conference

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
3

Implications and Questions for Computer Simulation

- Java has a number of features that have the potential to dramatically change the process used to **plan, build, execute, maintain, and distribute** simulation models
- **Important questions**
  - What are the benefits of Java and its related technologies for modelling and computer simulation?
  - Are there any disadvantages?
  - What is the state of the art of Java-based simulation?
  - What are the future prospects?
  - What shall we do with our existing simulation programs?

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
4

Overview of Java

Basic properties

- object-oriented
- familiar and simple
- safer and more reliable
- secure
- multithreaded
- interpreted and portable

⇒ **Programming language of the Web**

The Java Development Process

Java releases from Sun Microsystems

- January 1996

  JDK 1.0 for Solaris and Windows 95/NT

- May 1996

  JDK 1.0.2 for Solaris, Windows 95/NT, and Mac (3.66 MB)

- February 1997

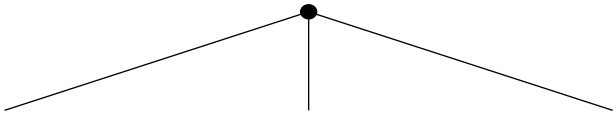
  JDK 1.1 (11.8 MB)

- December 1998

  JDK 1.2 (35.5 MB)

The Java Application Programming Interface (API)

Java libraries



Core libraries

Class libraries that every JDK must support

JDK 1.0: **8** core libraries
JDK 1.1: **22** core libraries
JDK 1.2: over **50** core libraries

Standard extensions to the JDK

Optional class libraries generally distributed with the JDK

APIs for specific hardware or application areas

Not necessarily intended to be distributed with the JDK

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
7

Why is Java so Attractive?

- Object-orientation
- Powerful API
- Similarity to C/C++
- Platform-independence
- Close relationship to other key technologies
- Applet concept
- Available for free

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
8

Java-Based Simulation

Important areas

- Online simulation and animation
- Simulation support packages
- Distributed simulation

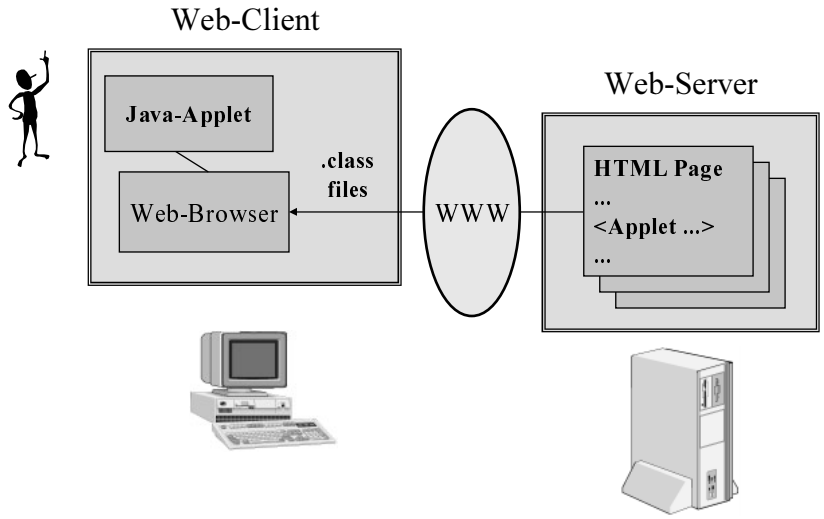
Additional information

http://ms.ie.org/websim/survey/survey.html

Online Simulation and Animation

- Simulations that are on the Web and available for execution
- Realisation alternatives
  - Client-side execution
    - Java applets
  - Server-side execution
    - CGI scripts
    - Java servlets

Client-Side Execution Using Java Applets



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
11

Java Applets

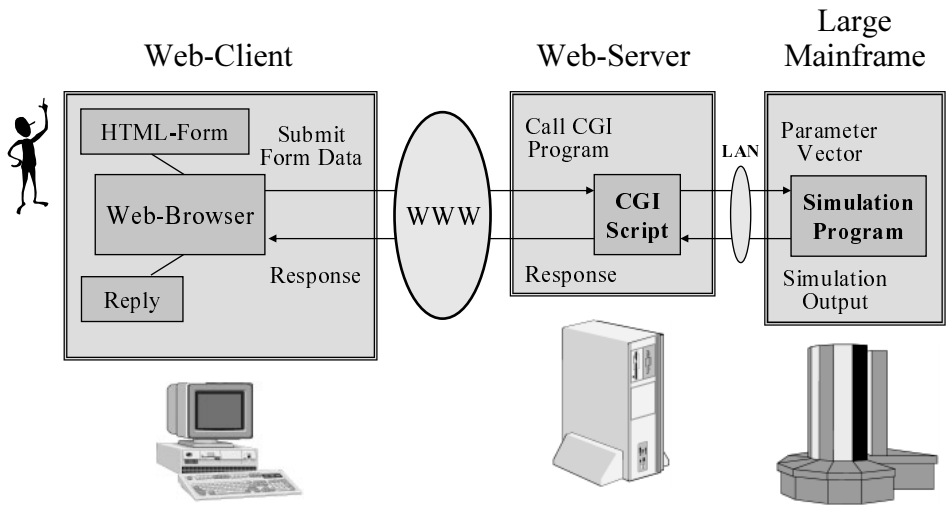
- **Advantages**
  - immediately executable anywhere in the World at any time on any platform without having to install anything
- **Disadvantages**
  - high network traffic
  - (re-)implementation in Java
  - extensive simulation runs require powerful clients
  - execution results may vary according to the applied browser
  - restricted access to the client system

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
12

Java Applets

- **Suitable for**
  - small- and medium-sized animations and simulations
- **Application fields**
  - teaching and education
  - demonstration of research results
  - sales promotion

Server-Side Execution Using CGI Scripts

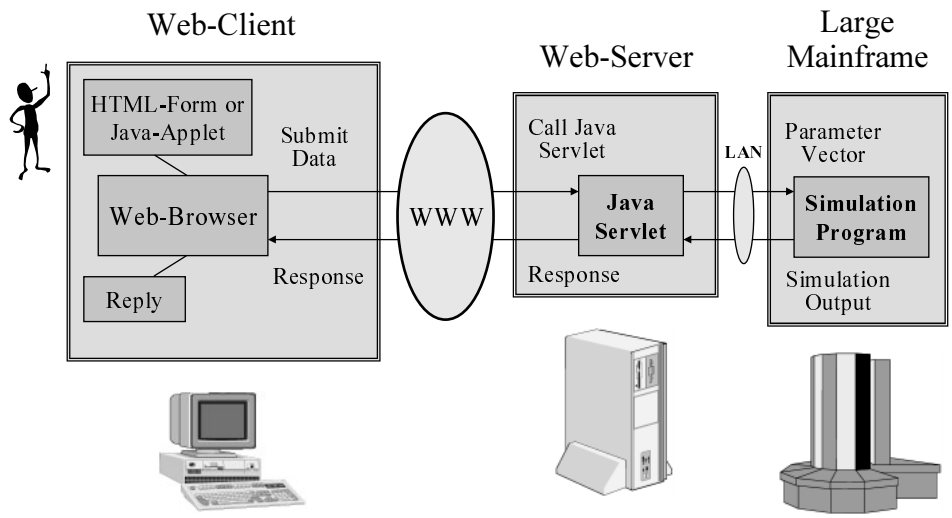


CGI-Scripts

- **Advantages**
  - thin clients
  - no reimplementation of existing simulation programs
  - small network traffic

- **Disadvantages**
  - no real-time animations („live“ simulations)
  - highly platform-dependent
  - difficult development, implementation, and fault detection
  - complex control of resource contention and consumption

Server-Side Execution Using Java Servlets



Java Servlets

- **Advantages**
  - thin clients
  - no reimplementation of existing simulation programs
  - small network traffic
  - platform-independence
  - much better performance and management characteristics than CGI scripts
  - simple implementation and usage
- **Disadvantages**
  - servlet capable Web-server

Simulation Support Packages

- Simjava
- Silk
- JavaSim
- JSIM
- DEVS-JAVA
- DESMO-J
- ...

Simjava

- Process based discrete event simulation package for Java
- Based on SIM++ (a discrete event simulation library for C++)
- Developed and distributed by

  The Institute for Computing Systems Architecture,

  Division of Informatics,

  University of Edinburgh.
- Freely available in full source code for download at

  http://www.dcs.ed.ac.uk/home/hase/simjava/

Simjava Packages

Simjava consists of three separate java packages

- simjava

  discrete event simulation library
- simanim

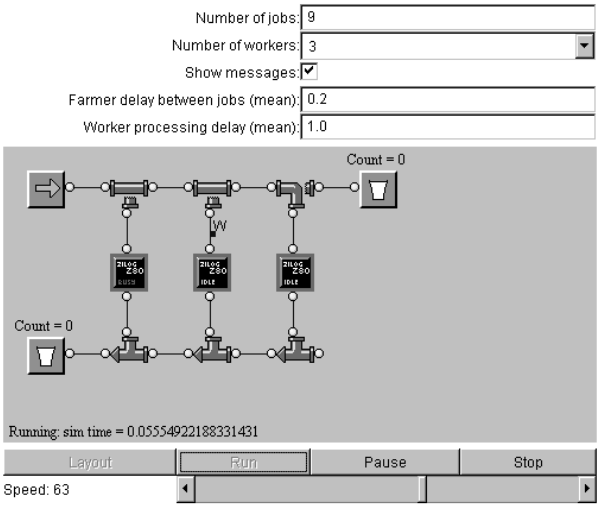
  package to provide animation
- simdiag

  collection of classes for displaying simulation results

⇒ **basic elements necessary for building animated discrete event simulations in Java**

Simjava Example

A simple work farm



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

21

Performance Comparison

simulation program

SIM++
C++ libraries

Simjava
Java libraries

Platform	Average execution time over 5 runs
C++ version under Solaris on Sun SPARCstation 5	**1538 ms**
Java application under Solaris on Sun SPARCstation 5	**12910 ms**
Java applet under Solaris/Netscape on Sun SPARCstation 5	**11214 ms**
Java applet under Windows NT/Netscape on Pentium 133 MHz	**9341 ms**

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

22

Distributed Simulation

- **Basic communication requirements**
  - remote procedure call (RPC)
  - remote method invocation (RMI)
    - java.rmi
- **Infrastructures for object-oriented distributed computing**
  - component frameworks
    - ActiveX/(D)COM
    - Corba
    - Java Beans
  - agent platforms
    - Voyager
    - Aglets

Distributed Simulation Infrastructures and Architectures

- Aggregate Level Simulation Protocol (ALSP)
- Distributed Interactive Simulation (DIS) Protocol
- High Level Architecture (HLA)
  - Developed by the US Department of Defense under the leadership of the Defense Modeling and Simulation Office (DMSO)
  - Goal: Support of simulation reuse and interoperability
  - Detailed information at: http://hla.dmso.mil/

Conclusions

Benefits of Java

- object-orientation
- standardised libraries for all mainstream computing
- support for standardised components
- support for distributed processing
- support for multi-threaded programming
- platform independence
- Applet concept
- Close relationship to other key technologies

Conclusions

What has to be improved?

- execution speed
  - Just-In-Time Compiler
  - Java Processor
  - Java HotSpot Performance Engine
  - trend towards faster and cheaper hardware
- stability of the core API

Zukunftstechnologien in der Modellierung und Simulation

✓ **Java/Web**

- Das Java-Phänomen
- Java-basierte Simulation
  - Online-Simulation and -Animation
  - Java-basierte Simulationspackete
  - Verteilte Simulation

• **Komponenten**

- High Level Architecture (HLA)
  - Überblick
  - Beispiel

Gliederung

- Motivation
- Grundlagen
- Beispiel
- Offene Probleme
- Ausblick

Motivation

Nachteile monolithischer Simulationsmodelle

- festgelegt auf genau eine Modellierungstechnik
- plattform-abhängig
- teuer in der Erstellung und Ausführung
- schwer überschaubar
- schwierig zu warten und zu erweitern
- nur schwer auf andere Problemstellungen übertragbar

Motivation

Heute zunehmend gefordert

- Verteilung
- Wiederverwendbarkeit
- Interoperabilität

⇒ **Komponenten-basierte Simulation**

HLA - High Level Architecture für die Modellierung und Simulation

Software-Architektur zur Entwicklung Komponenten-basierter Simulationen

- im September 1996 vom DoD (US Department of Defense) zum verbindlichen Standard für alle künftigen Neuentwicklungen von Simulationen im militärischen Bereich erklärt
- gleichzeitig zur freien Nutzung im zivilen Bereich offengelegt

Hintergrund

- Das US Verteidigungsministerium hat weltweit mit Abstand das größte Auftragsvolumen an Simulationen.
- Grund: Simulationen ermöglichen es, teure Waffensysteme zu erproben, ohne diese konkret realisieren zu müssen („try before buy").
- Anfang der 90er Jahre ist die Zahl vorhandener Simulationen derart angewachsen, daß der Überblick verlorenging.

Hintergrund

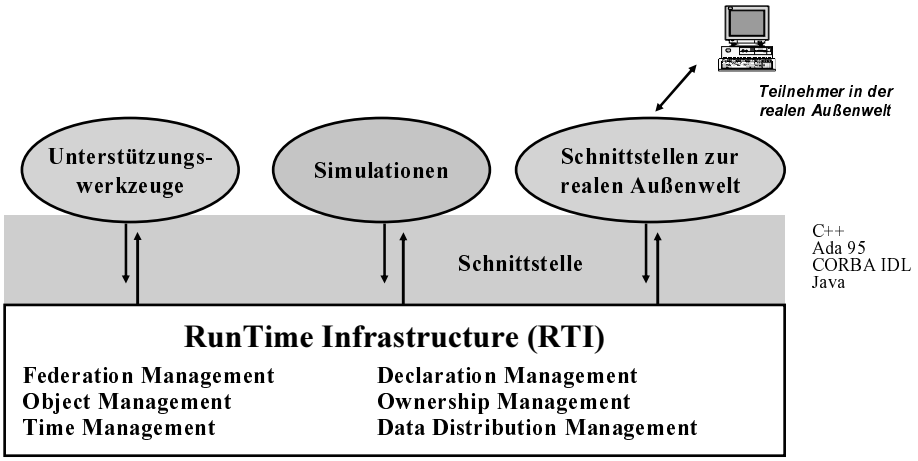
- Das Ende des kalten Krieges und die damit verbundenen drastischen Einsparungen im Militärhaushalt zwangen zu einem radikalen Umdenken hinsichtlich kostengünstiger und wiederverwendbarer Simulationen.

➢ **Geburtsstunde der HLA**

Überblick über die HLA

- Grundidee der HLA
  - strikte Trennung von Simulationsfunktionalität und Infrastruktur für die Interoperabilität
- Grundelemente einer HLA-Föderation
  - Föderierte (Federate)
  - RunTime Infrastructure (RTI)
- Kommunikation zwischen Simulationsteilnehmern
  - ausschließlich über die RTI

Funktionale Übersicht über die HLA



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
35

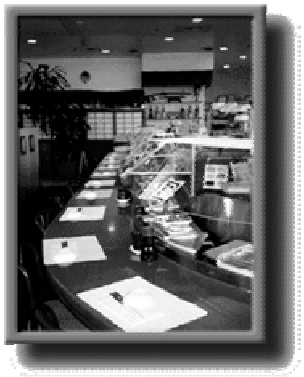
Weitere integrale Bestandteile der HLA

- **Object Model Template (OMT)**
  - Objektklassen
  - Interaktionsklassen
- **HLA Regelwerk**
  - Föderationsregeln (Regeln 1 – 5)
  - Föderiertenregeln (Regeln 6 – 10)

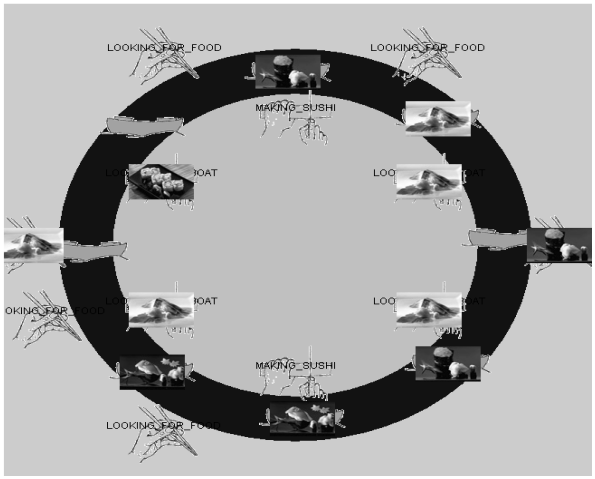
Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
36

Beispiel

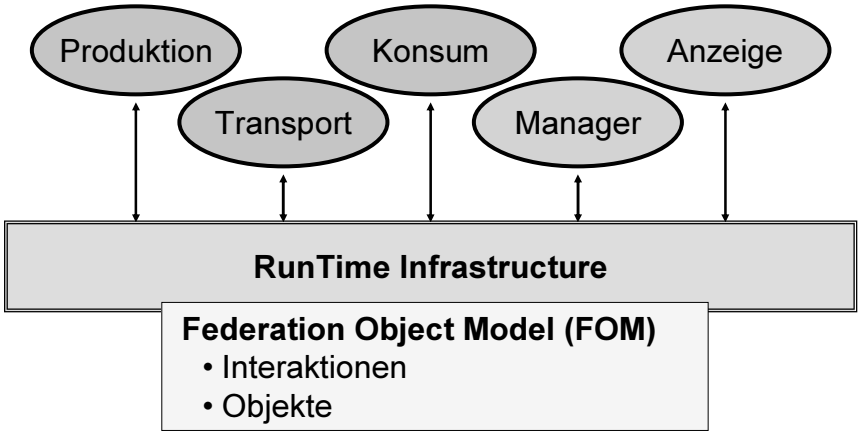
Die Sushi-Restaurant Föderation



Gesamtübersicht

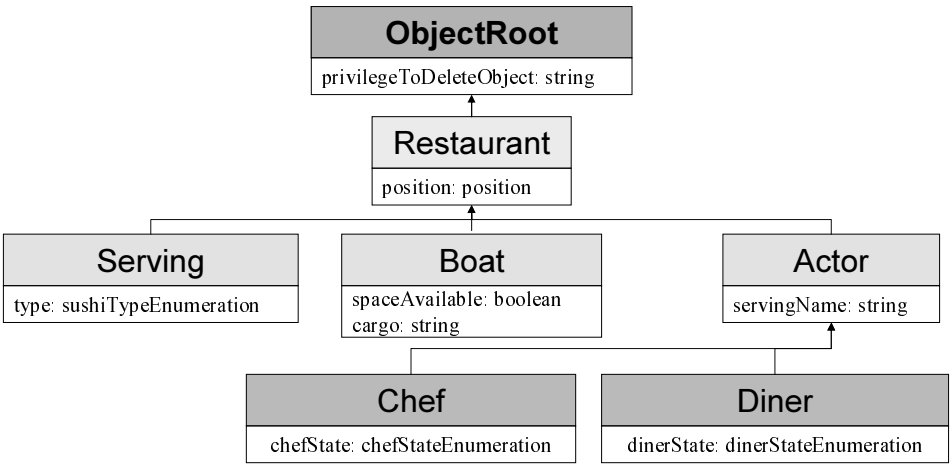


Beteiligte Komponenten



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
39

Objektklassen des gemeinsamen Objektmodells



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
40

Verwaltung gemeinsamer Daten

Zuständige RTI-Dienste

- Object Management

  Erzeugen und Löschen von Objektinstanzen

- Declaration Management

  Abonnement von Objektattributen

- Ownership Management

  Übertragung von Zugriffsrechten auf Objektattribute

- Data Distribution Management

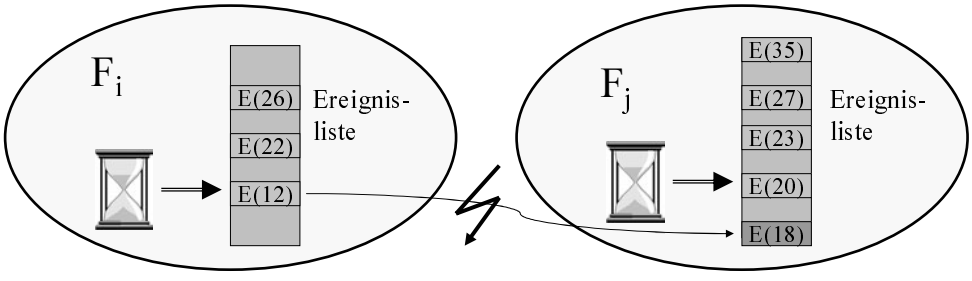
  Effizientes Routing von Daten

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
41

Zeitmanagement

Hauptproblem: Synchronisierung der Ereignisausführungen, so daß

- die Kausalität gewahrt bleibt
- möglichst wenig Synchronisationsaufwand entsteht
- möglichst viele Ereignisse konkurrent ausgeführt werden



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
42

Zeitmanagement

Der **Zeitmanagementdienst** der RTI unterstützt folgende Synchronisationsmechanismen:

- **konservative Synchronisation**
- **optimistische Synchronisation**
- **keinerlei Synchronisation**

Zeitmanagement

Für jeden Föderierten muß der Grad der Teilnahme am Zeitmanagement festgelegt werden:

zeitreguliert \ *zeitregulierend*	ja	nein
ja	strenge Zeitsynchronisation (Produktion, Konsum, Transport)	Erfassung und Anzeige von Simulationsdaten (Anzeige)
nein	zeitlicher Schrittgeber (Manager)	keinerlei Zeitsynchronisation

Der Prozeß

1

Der Benutzer startet die RTI

RTIExec

2

Der Benutzer startet einen Föderierten

Federate 1

Der Föderierte erzeugt die Föderation

FedExec

Zuerst wird die RTI gestartet.
Ein Föderierter, der als Manager agiert, erzeugt eine Föderation mit Hilfe des RTI Dienstes "createFederationExecution".

Die RTI reserviert einen Namen für die Föderation.

Nachdem eine Föderation existiert, können sich weitere Föderierte anschließen.

Weitere Föderierte schließen sich der Föderation an

Federate 2

Federate 3

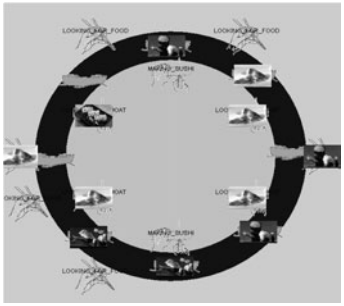
3

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

45

Vorführung

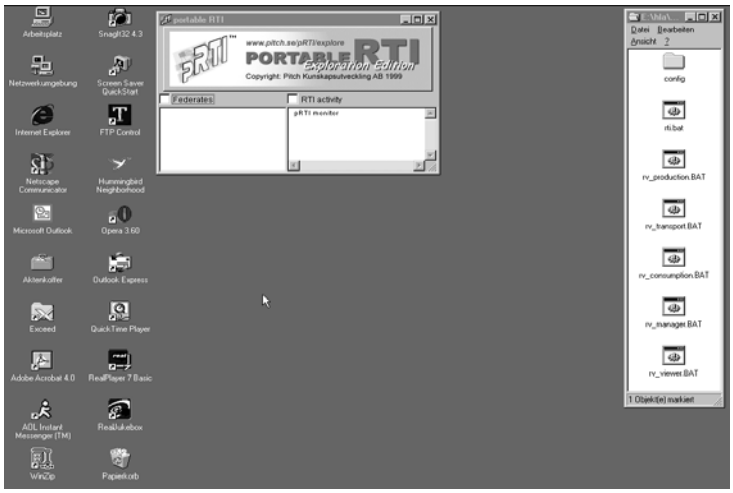
Komponenten-basiertes Simulationsmodell
der Sushi-Restaurant Föderation



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

46

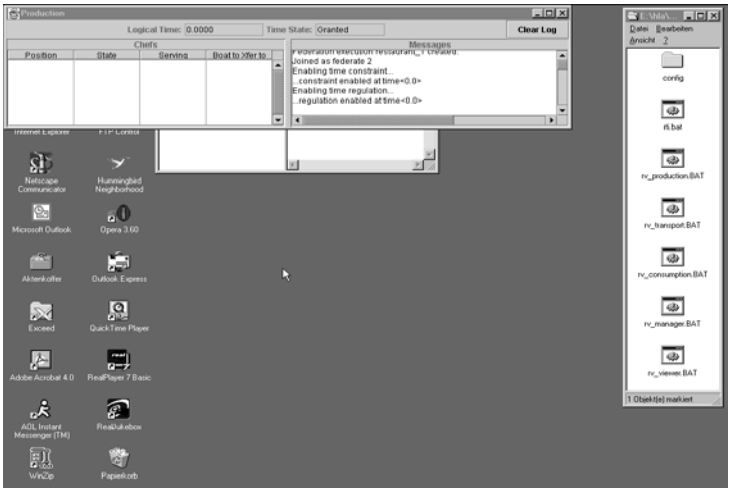
Start der RunTime Infrastructure (RTI)



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

47

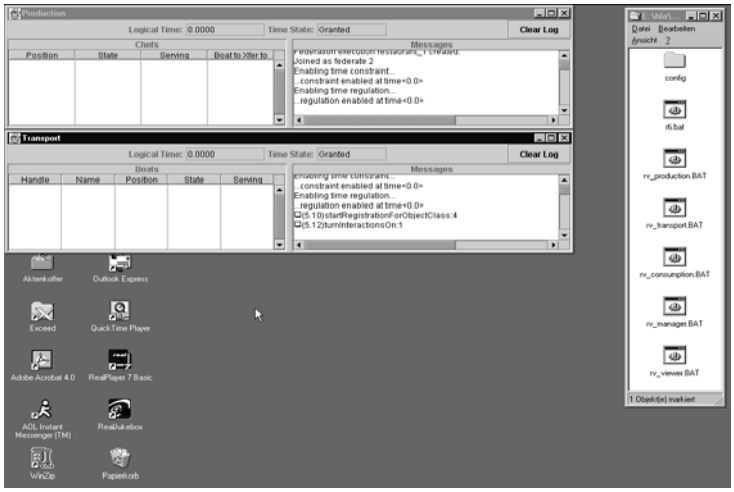
Start des Produktions-Föderierten



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

48

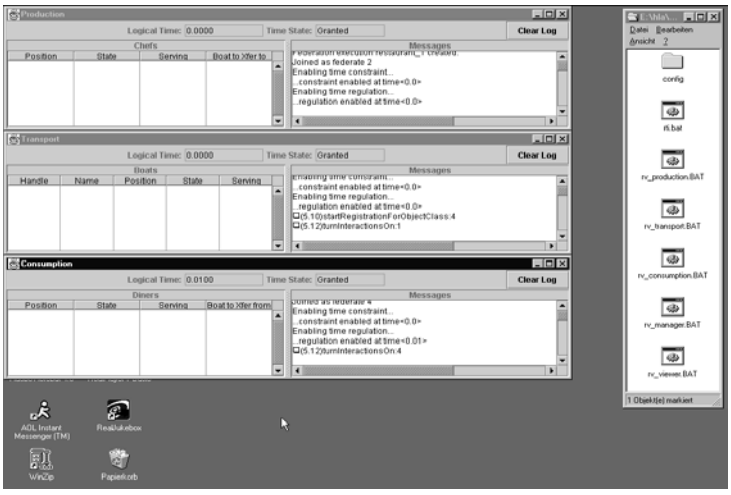
Start des Transport-Föderierten



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

49

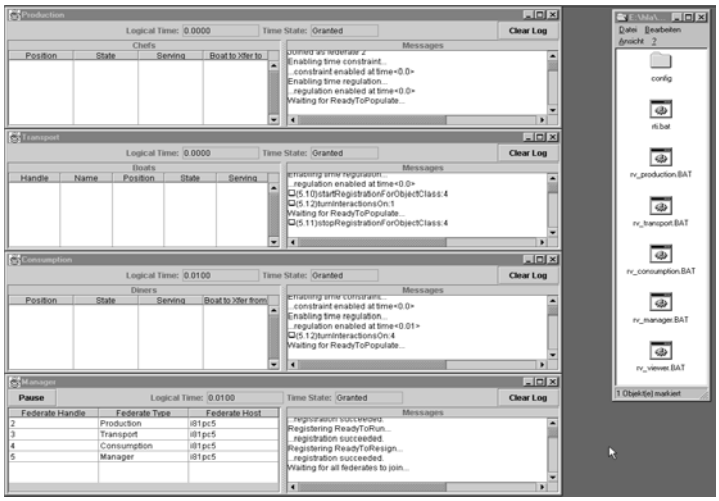
Start des Konsum-Föderierten



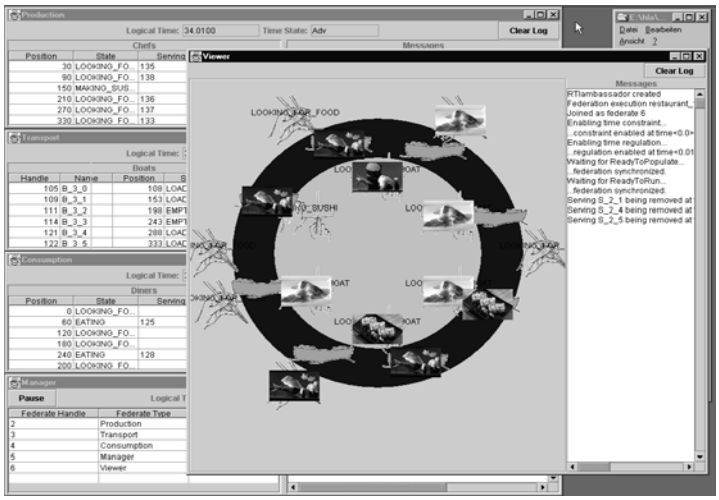
Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)

50

Start des Manager-Föderierten



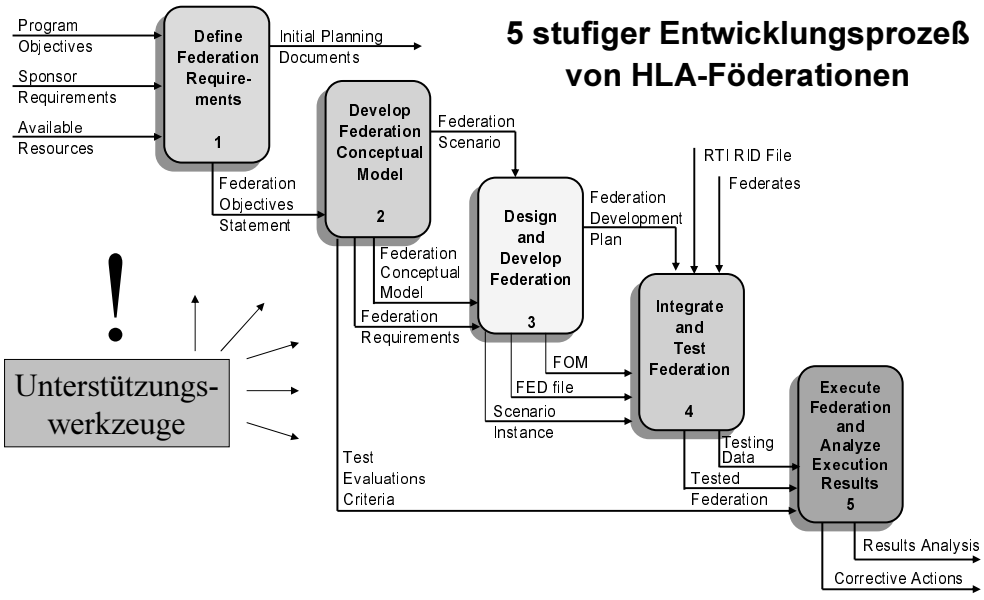
Start des Anzeige-Föderierten



Offene Probleme der HLA

- Viele etablierte Modellierungswerkzeuge sind noch nicht HLA-konform.
- Der professionelle Einsatz der HLA erfordert Weiterentwicklung der RTI.
- Um eine breite Akzeptanz zu erreichen, müssen auch Anwendungsfelder im zivilen Bereich erschlossen werden.
- Der Aufbau komplexer Föderationen erfordert Unterstützungswerkzeuge für die einzelnen Entwicklungsstufen wie z.B.
  - Verfahren zur Modellpartitionierung und Ressourcenzuordnung
  - Methoden zur Leistungsbewertung und zum Tuning verteilter Komponenten-basierter Simulationen
  - Verifikations- und Validationsmethoden

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
53



Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow)
54