

SIMPLEX3

Funktionsvielfalt

Vielseitigkeit

**Benutzer-
freundlichkeit**

Neue Werkzeuge

**Graphischer
Modellaufbau**

**Konfigurierbare
Bedienoberfläche**

Animation

SIMPLEX3

Positionierung des Simulationssystems und der Werkzeuge

Simulatortypen	Anwender	System/Werkzeug
Selbst programmierte Simulationsmodelle	Informatiker/Programmierer Simulationsexperte	
Sprachgestützte Simulatoren	Simulationsexperte	SIMPLEX3
Systemnahe Simulatoren		
Baustein Simulatoren	Simulationsexperte Anwender	Modelleditor
Parametrisierbare Simulatoren	Anwender	Konfigurierbare Modelloberfläche

SIMPLEX3

Unterstützung in allen Phasen der Simulation

Modellerstellung Modellverwaltung

Modellbeschreibungssprache
Simplex-MDL

C++ - Schnittstelle

Modellaufbau mit
Text- bzw. Graphikeditoren

Anlegen von Modellbanken

Versionenkonzept

Vorgefertigte Modellbanken

Experimente Experimentdaten- & Laufverwaltung

Komfortable
Experimentierumgebung

Trennung von Modell
und Experiment

Flexible und einfache
Datenaufzeichnung

Experimentbeschreibungs-
sprache EDL

Optimierung

Auswertung Präsentation

Präsentationsgraphiken

Statistische Auswertung

Datenexport über die
offene C++ - Schnittstelle

SIMPLEX3

Vielseitigkeit durch Simplex MDL

Deklarative Modellbeschreibungssprache mit systemnahen Konzepten und Sprachkonstrukten auf systemtheoretischer Grundlage

Simplex MDL bietet eine formale, nicht prozedurale Modellspezifikation



Modellklassen

- Zeitkontinuierliche Modelle
- Ereignisorientierte Modelle
- Auftragsorientierte Modelle
- Transportmodelle
- kombinierte Modelle

SIMPLEX3

Struktur einer Basiskomponente

Die Basiskomponente als Grundbaustein in Simplex MDL

Basiskomponente

Name: Komponente1
Deklaration: <ul style="list-style-type: none"> • Modellgrößen • Zufallsgrößen • Indikatoren • Locations • mobile Komponenten • Einheiten
Dynamik: <ul style="list-style-type: none"> • algebraische Gleichungen • Differentialgleichungen • Ereignisse • Fallunterscheidungen

Der Name identifiziert die Komponente

Im Deklarationsteil werden die Modellelemente und ihre Eigenschaften angegeben

Die Dynamikbeschreibung spezifiziert, unabhängig von der Reihenfolge, das Verhalten der Komponente

```

BASIC COMPONENT QueueK
MOBILE SUBCOMPONENTS
    OF CLASS CustomerK
USE OF UNITS

DECLARATION OF ELEMENTS
    STATE VARIABLES
    DEPENDENT VARIABLES
    RANDOM VARIABLES
    TRANSITION INDICATORS
    LOCATIONS

DYNAMIC BEHAVIOUR
    DIFFERENTIAL EQUATIONS
END OF QueueK
    
```

SIMPLEX3

Modellgrößen in Simplex MDL

Die Modellgrößen umfassen die Eigenschaften (Attribute) einer Komponente

Basiskomponente

Name: Komponente1

Deklaration:

- Modellgrößen
- Zufallsgrößen
- Indikatoren
- Locations
- mobile Komponenten
- Einheiten

Dynamik:

- algebraische Gleichungen
- Differentialgleichungen
- Ereignisse
- Fallunterscheidungen

Modellgrößen

Zustandsvariablen

- Ereignisse
- Differentialgleichungen

Abhängige Variablen

- algebraische Gleichungen

Sensorvariablen

- Verbindungen
- Hierarchischer Modellaufbau

Beispiel: Deklaration von Modellgrößen

```

DECLARATION OF ELEMENTS
STATE VARIABLES
DISCRETE
  TAnkunft (REAL[s]) := 0 [s]
CONTINUOUS
  Fuellstand (REAL[Liter]):=0 [Liter]
DEPENDENT VARIABLES
CONTINUOUS
  Rate (REAL[Liter/s]):=0.0 [Liter/s]

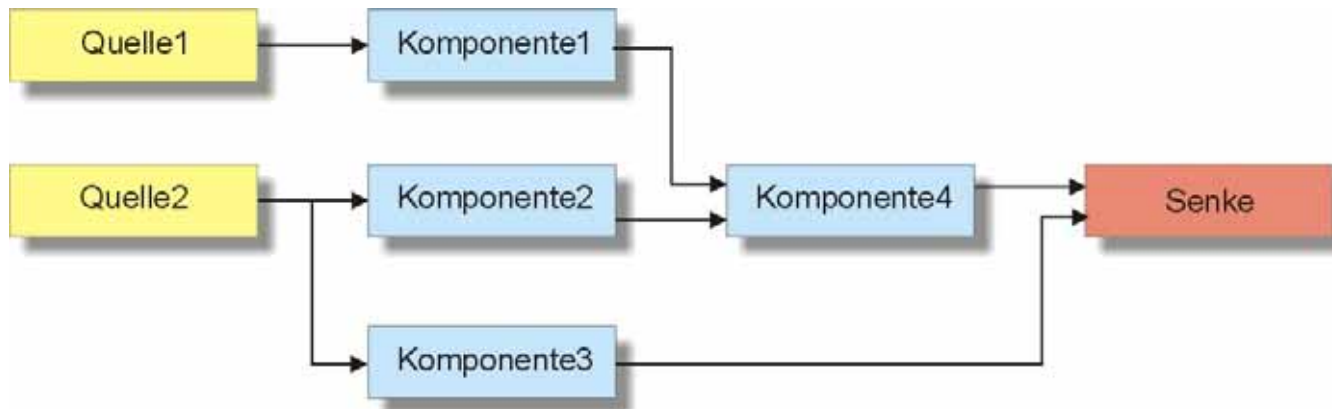
```

SIMPLEX3

Komponentenorientierter Ansatz

Jedes Modell setzt sich aus ein oder mehreren Komponenten zusammen, die durch Verbindungen miteinander verknüpft werden können

Modellaufbau mit Hilfe von Basiskomponenten und Verbindungen

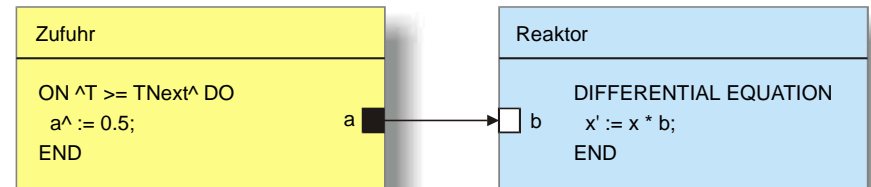


Verbindungen

Kausale Abhängigkeiten zwischen Komponenten

Glass Box Konzept

Spezifikation in High Level Komponenten



Zufuhr.a --> Reaktor.b

SIMPLEX3

Hierarchischer Modellaufbau

Strukturierung und schrittweise Verfeinerung von Simulationsmodellen

Der hierarchische Modelaufbau

Verfeinerung der Abstraktionsebene

Beliebig viele Abstraktionsebenen

Voraussetzungen

Untergeordnete Komponenten bestehen wieder aus Komponenten und Verbindungen

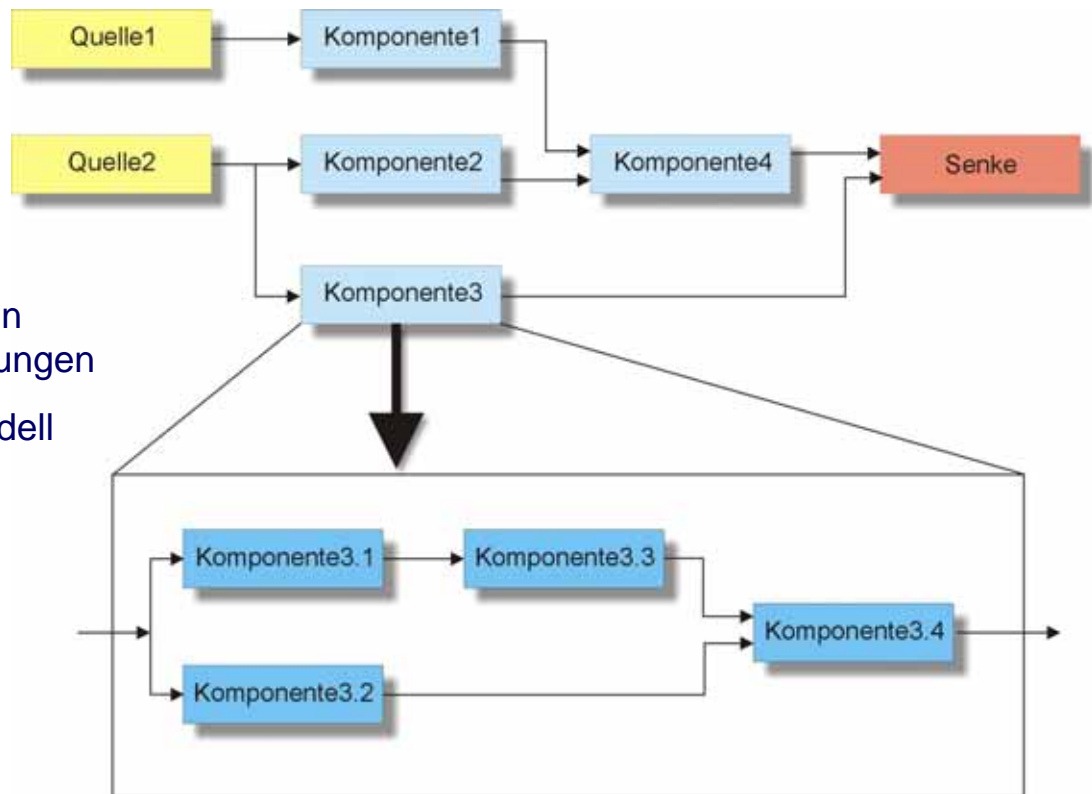
Schnittstellen zum übergeordneten Modell müssen erhalten bleiben

Vorteile

Übersichtliche Strukturierung

Systemnahe Modellierung

Rapid Prototyping



SIMPLEX3

Ein benutzerfreundliches Simulationssystem

Leicht zu bedienen - leicht zu erlernen - kostengünstig - universell einsetzbar

Bedienung

Graphische Bedienoberfläche
Kontextsensitive Bedienfunktionen

Dokumentation

Benutzerhandbuch

- Umgang mit dem Simulationssystem
- Simulationslehrbuch
- in mehreren Sprachen erhältlich

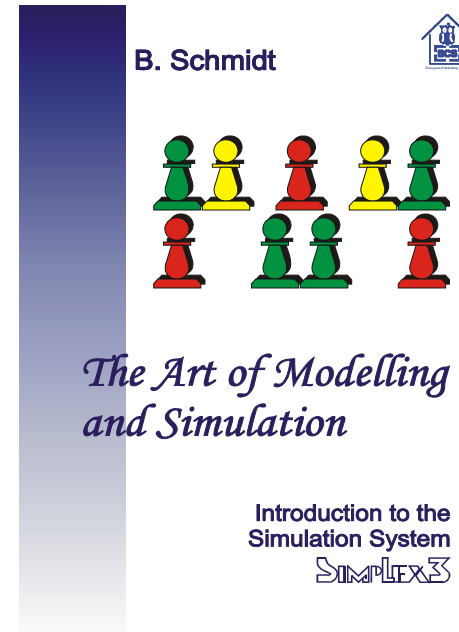
Referenzhandbücher

Kosten

Simulationssystem Simplex3: 0 €
Referenzhandbücher: 0 €
Benutzerhandbuch: 48 € / 75 €

Einsatzmöglichkeiten

In allen Bereichen der Simulation
Für alle Anwendergruppen geeignet
International: mehrsprachig



B. Schmidt: The Art of Modelling and Simulation

Introduction to the Simulation System Simplex3
SCS-Europe Publishing House, Ghent 2000
ISBN 1-56555-228-8
ISBN 3-936150-06-0

Neue Werkzeuge

Der Modelleditor: ein Layouteditor zum graphischen Modellaufbau

Übersicht

Graphischer Editor zum Erstellen von Simulationsmodellen

Anlegen von Bausteinen (Blocks) beliebiger Art

Wiederverwendbarkeit der Bausteine durch das Erzeugen von Vorlagen

Unabhängig vom Simulationssystem

Adaptionsfähig an beliebige komponentenorientierte Simulationssysteme

Entwickelt in Java nach dem MVC-Pattern

Funktionsumfang

Erstellen von graphischen Repräsentationen (Blöcke) auf Grundlage von Komponenten

Erstellen von Blockvorlagen

Erstellen von Verbindungen zwischen den Blocks (kausale Abhängigkeiten)

Konsistenzprüfung der Verbindungen

Bearbeitungsfunktionen für das entstandene Layout

Speichern des Layouts

Erzeugen des Codes für das Simulationsmodell

Neue Werkzeuge

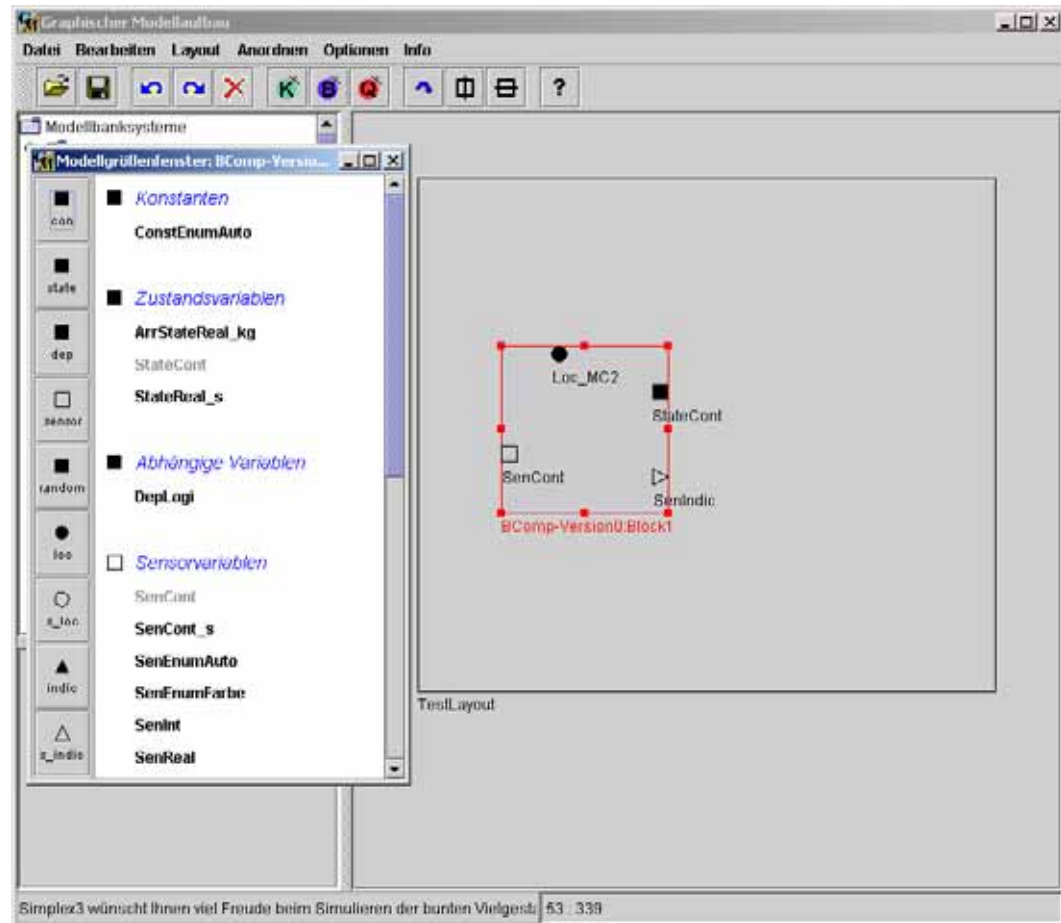
Der Modelleditor: ein Layouteditor zum graphischen Modellaufbau

Anlegen eines Blocks

Anzeigen des Blocks
als Rechteck

Anzeigen der Modellgrößen

Einfügen der Modellgrößen in
das Layout mit Drag&Drop



Neue Werkzeuge

Der Modelleditor: ein Layouteditor zum graphischen Modellaufbau

Verbindungen

Verbindungen zwischen Komponenten definieren kausale Abhängigkeiten

Definition von Anschlüssen für höhere Komponenten

Erzeugter Code für die höhere Komponente:

```

MDL - Code
HIGH LEVEL COMPONENT TestLayout

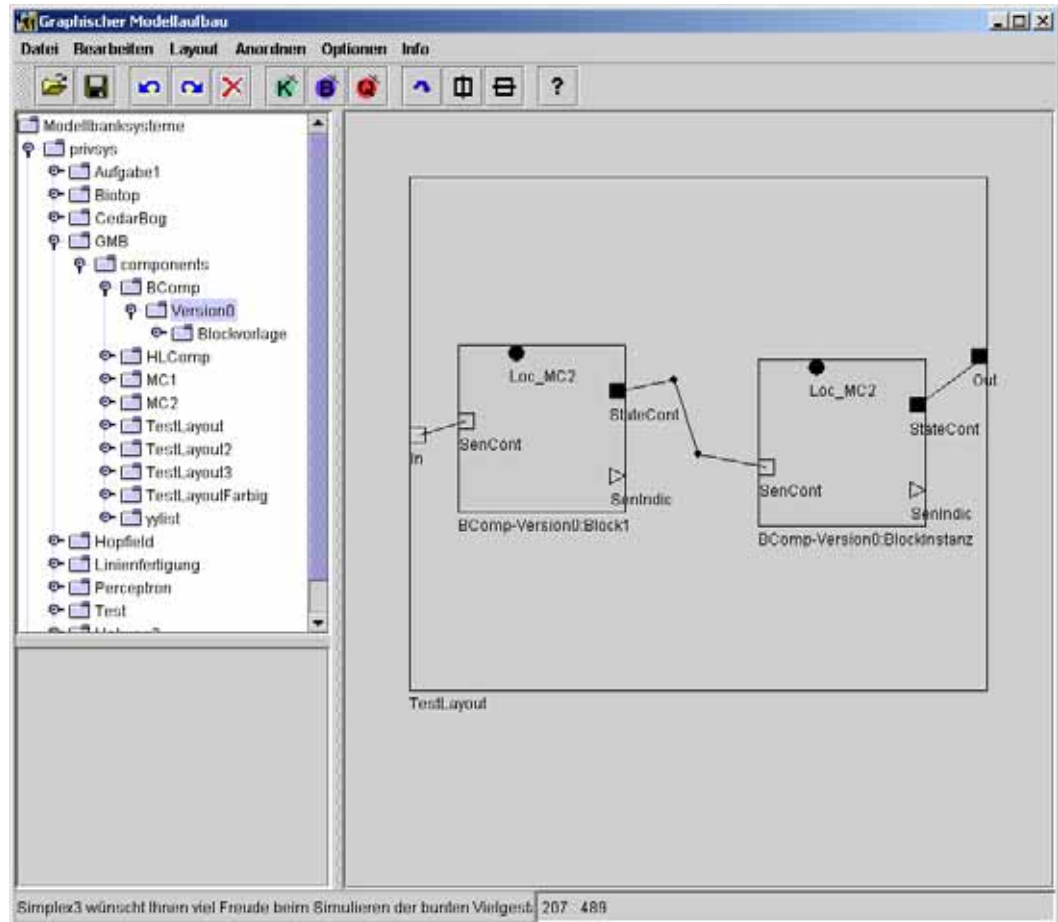
SUBCOMPONENTS
Block1 OF CLASS BComp,
BlockInstanz OF CLASS BComp

INPUT CONNECTIONS
In --> Block1.SenCont;

OUTPUT EQUIVALENCES
Out := BlockInstanz.StateCont;

COMPONENT CONNECTIONS
Block1.StateCont --> BlockInstanz.SenCont;

END OF TestLayout
    
```



Neue Werkzeuge

Die konfigurierbare Bedienoberfläche

Übersicht

Werkzeug zur Durchführung von Experimenten an Simulationsmodellen

Individuell konfigurierbar für alle Arten von Simulationsmodellen

Geeignet für den Anwender einer Simulation

Entwickelt in C++ für Modelle des Simulationssystems Simplex3

Konfigurierbarkeit

Systemnah gestaltbares Layout der Bedienoberfläche

- Parameterfenster
- Hintergrundbilder

Angabe der relevanten Simulationsdaten

- Simulationsparameter
- aufzuzeichnende Daten
- zu exportierende Daten

Einfach konfigurierbar über

- textbasierte Konfigurationsdateien
- Layouteditor

Neue Werkzeuge

Die konfigurierbare Bedienoberfläche

Funktionsumfang

- Parametrisierung des Modells
- Verwalten von Szenarios
- Durchführen von Simulationläufen
- Ergebnispräsentation
- Datenexport

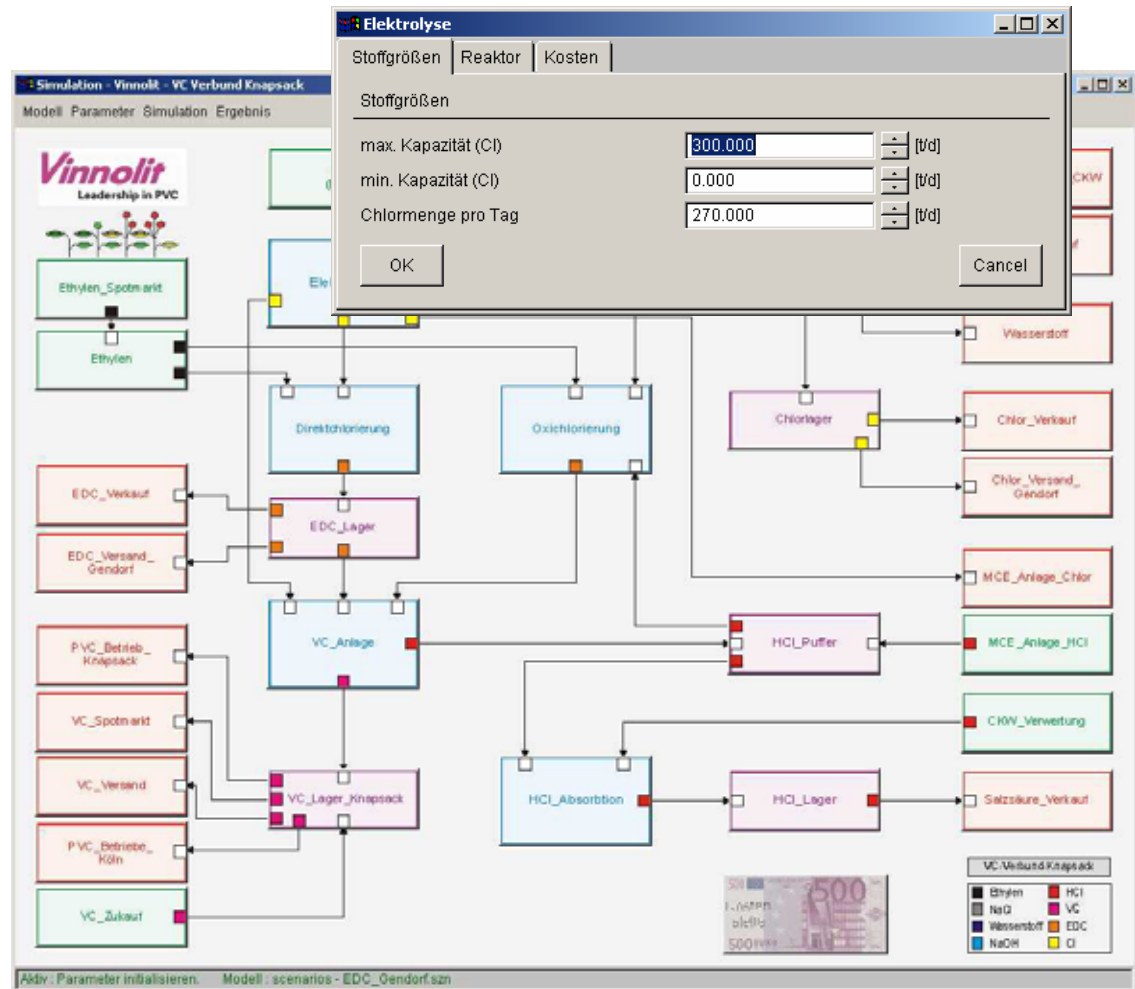
Beispiel:

Das Simulationsmodell VC-Verbund-Knapsack

Simulation einer Chemieanlage
inklusive der Kostenrechnung

Systemnaher,
Komponentenorientierter Aufbau

Kontinuierliches,
zustandsorientiertes Modell



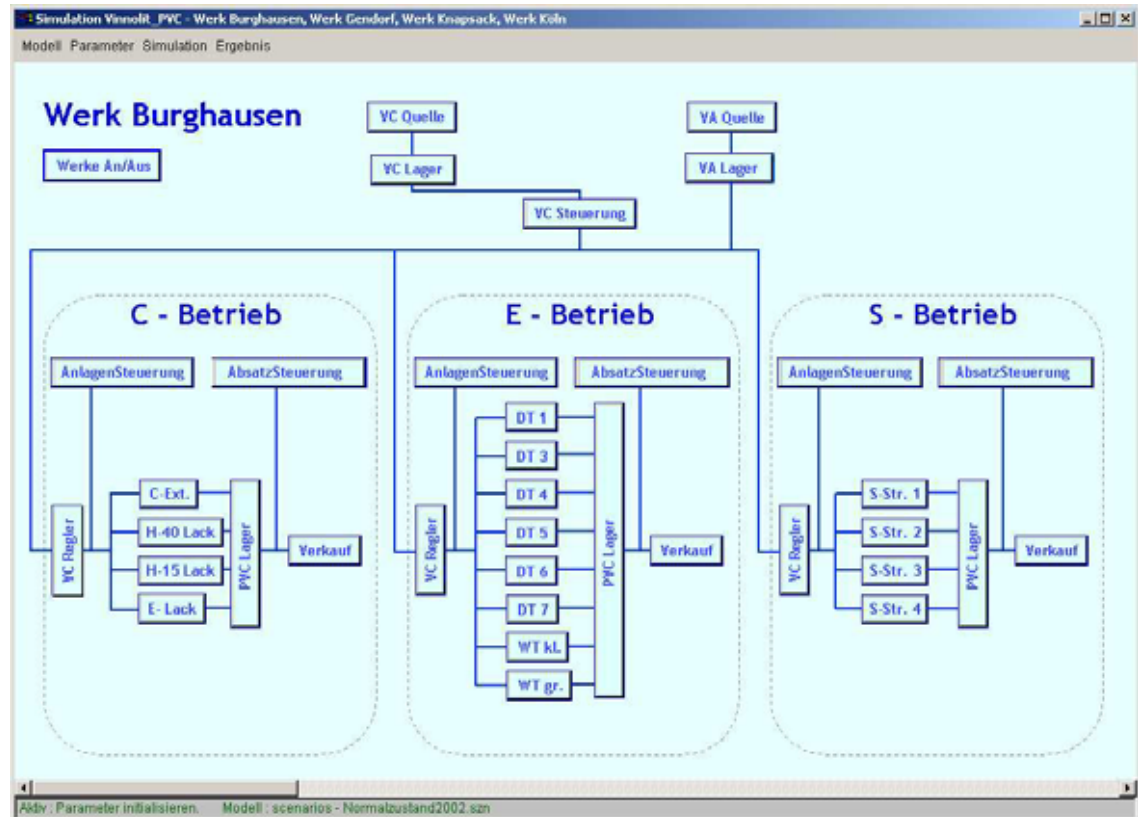
Neue Werkzeuge

Die konfigurierbare Bedienoberfläche

Beispiel:
Das Simulationsmodell Vinnolit - PVC

Simulation der Auftragseinlastung
in verschiedene Produktionsstrassen
von mehreren Betrieben

Auftragsorientiertes Modell
Gemischt kontinuierlich und diskret



Neue Werkzeuge

Das Animationsframework Anina3

Übersicht

Universelles und webfähiges Animationssystem

Freie Konfigurierbarkeit der Animationslayouts

Erweiterbare Graphikbibliothek

Adaptionsfähig an beliebige Simulationssysteme durch offene Schnittstellen und Datenformate

Animationskontrollen im System vorhanden, erweiterbar

Designsprache Java

Frei konfigurierbare Animationslayouts

Animationslayouts beliebig gestaltbar zur Animation aller Arten von Simulationen

Graphikobjekte aus Bibliothek oder frei programmierbar Java

Erstellung der Graphikobjekte in Java nutzt die Mächtigkeit der Programmiersprache

Graphikbibliothek

Erlaubt effiziente und schnelle Erstellung von Animationen

Wiederverwendbarkeit von selbst erstellten Graphikobjekten

Neue Werkzeuge

Das Animationsframework Anina3

Adaptionsfähigkeit durch offene Datenschnittstellen

Anbindung der Simulationsdaten von beliebigen

- Simulationssystemen
- selbstgeschriebenen Programmen

Implementation des Java-Datenquellen-Interfaces

Unterstützung von

- einfachen Datentypen
- Records
- Mobile Daten

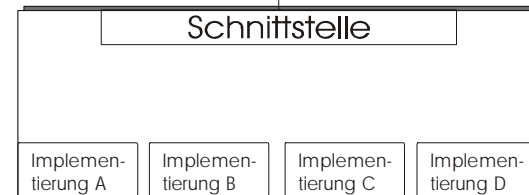
Erweiterbarkeit

Java-Datenquellen-Interfaces

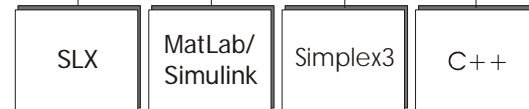
Applet



Datenquelle



Simulationssysteme



Client
Server

Neue Werkzeuge

Das Animationsframework Anina3

Webfähigkeit

Einsatz ist möglich als

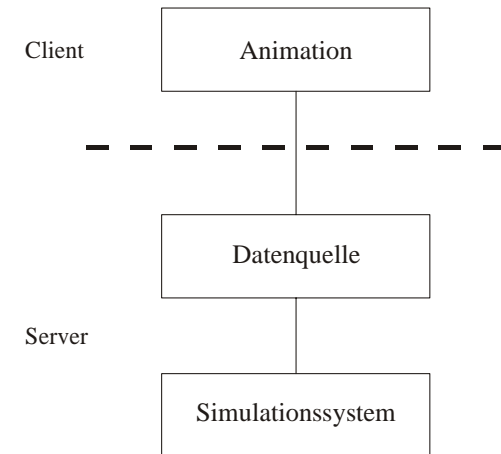
- Applet und als Applikation
- Client-Server Betrieb und stand alone

Client-Server Betrieb

- Client: Anzeigefunktionalität
- Server: Bereitstellen der Daten

Konzepte zur webbasierten Ausführung

- Datenpuffer
- minimale Datenübertragung
- Paralleles Anzeigen und Laden von Daten



Neue Werkzeuge

Das Animationsframework Anina3

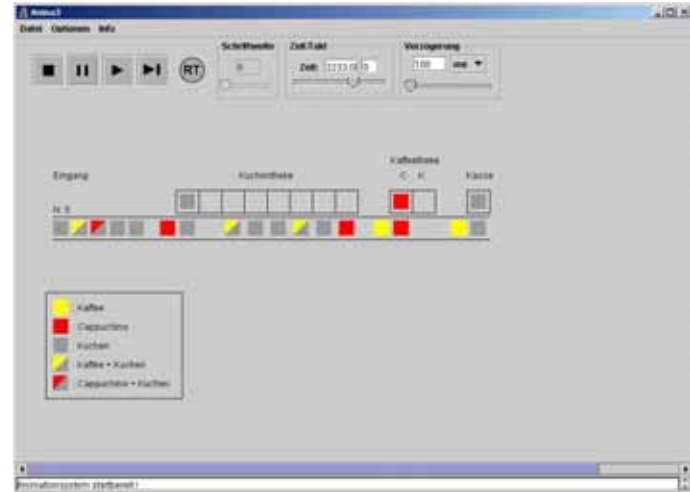
Beispiele: Anbindung von Anina3 an Simplex3

Animation des Simulationsmodells Cafeteria

Warteschlangenmodell

Animation von:

- Warteschlangen
- Bedienstationen
- Laufwegen
- Kunden



Animation des Simulationsmodells Adam

Grundlage Simulationsmodell: Referenzmodell PECS
(**P**hysis, **E**motion, **C**ognition, **S**ocial Status)

Visualisierung des PECS-Agenten Adam

- interner Zustand
- Entscheidungs- und Planungsprozesse
- Umwelt

