

Objektorientierte Simulation

Stefan Bodamer

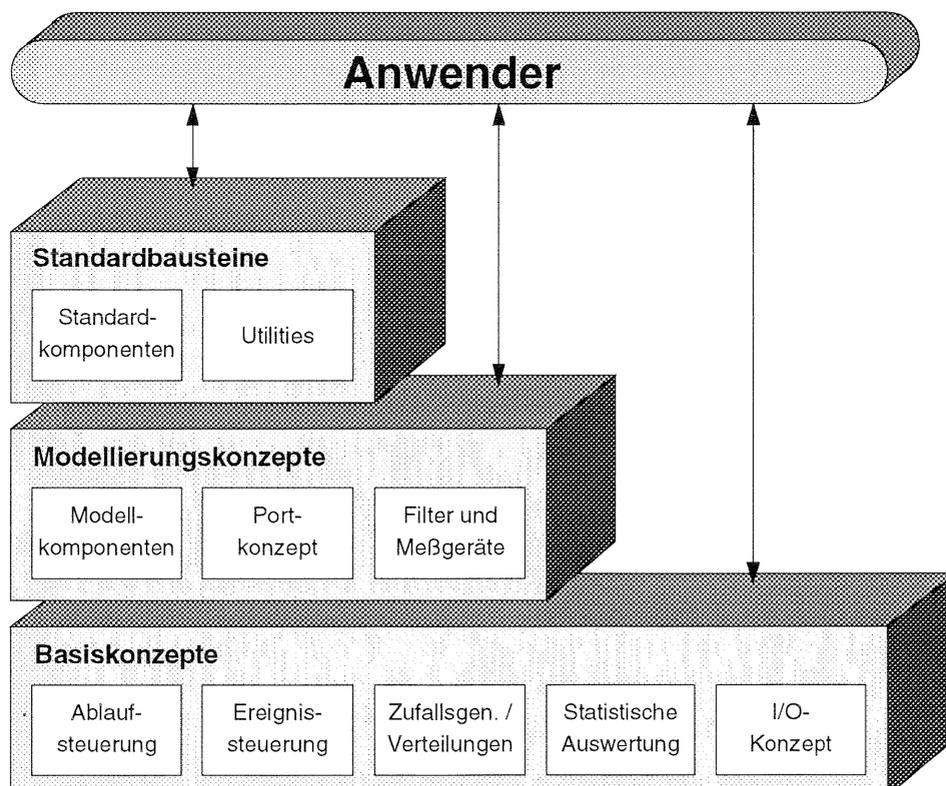
Gliederung

- **Übersicht über die Simulationsumgebung**
- **Modellierungskonzepte**
- **Meßwerterfassung**
- **Statistische Auswertung**
- **Ablaufsteuerung**
- **Zusammenfassung**

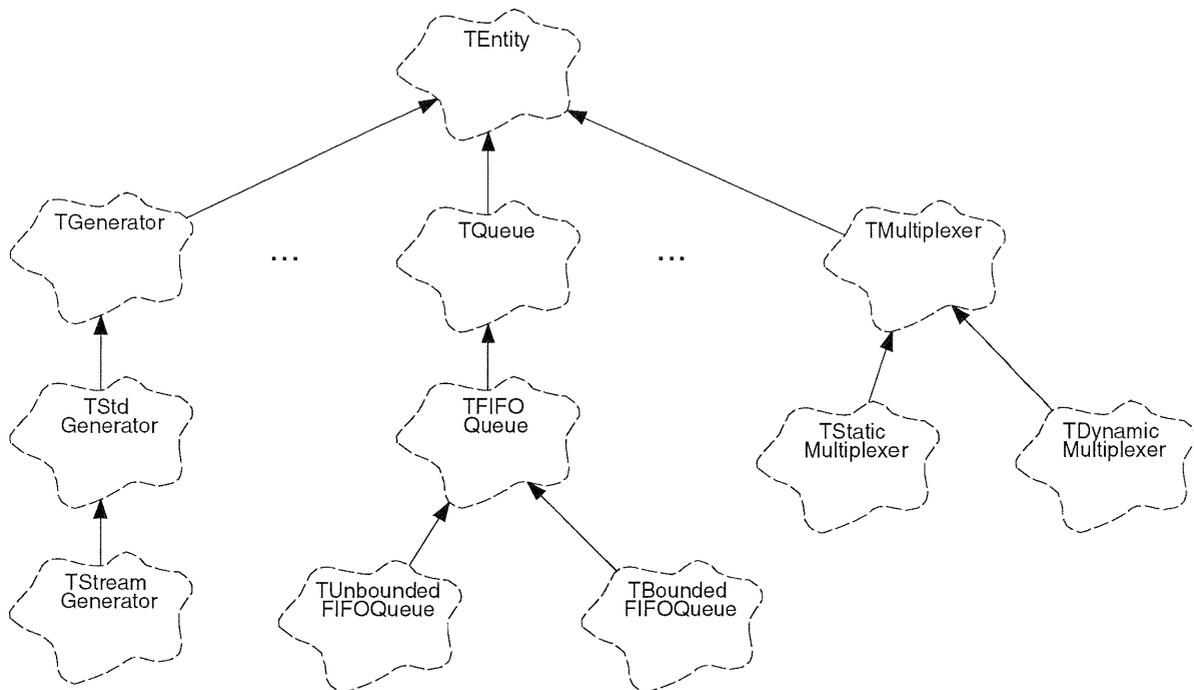
Eigenschaften der Simulationsumgebung

- **Objektorientierte Klassenbibliothek**
 - Implementierung in C++
 - Verwendung von Container-Klassen
- **Unterstützung bei**
 - Umsetzung des Modells
 - Durchführung der zeitdiskreten, ereignisgesteuerten Simulation
 - Erfassung und Ausgabe von Meßwerten

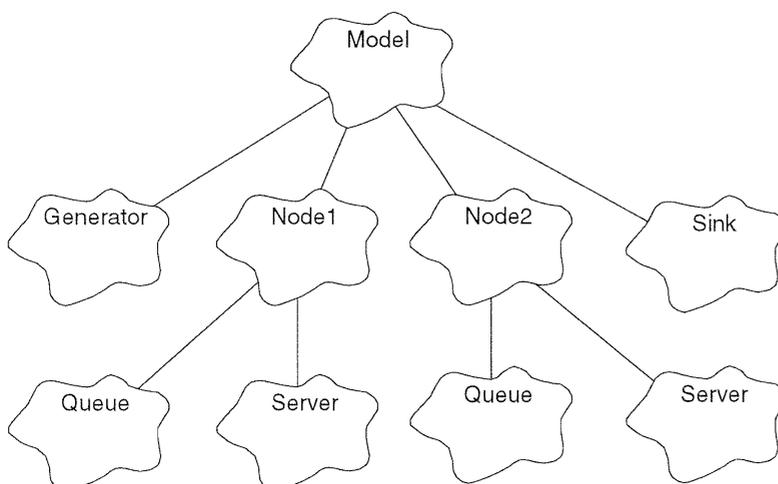
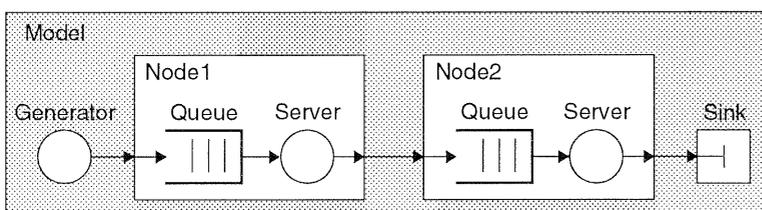
Struktur der Simulationsbibliothek



Klassenhierarchie bei Modellkomponenten



Objekthierarchie bei Modellkomponenten



Unterstützung der Dekomposition durch

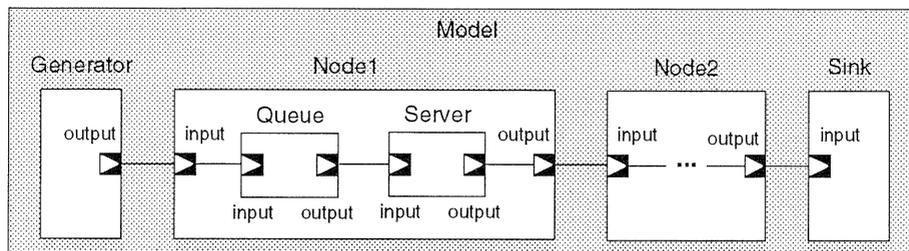
• Hierarchiebildung

- Verzeigerung („Owner“)
- Enthaltensein
- ⇒ Objektbaum

• Namenskonzept

- Identifizierung von Komponenten
- Zugriff über zentrale Verwaltungsinstanz

Portkonzept

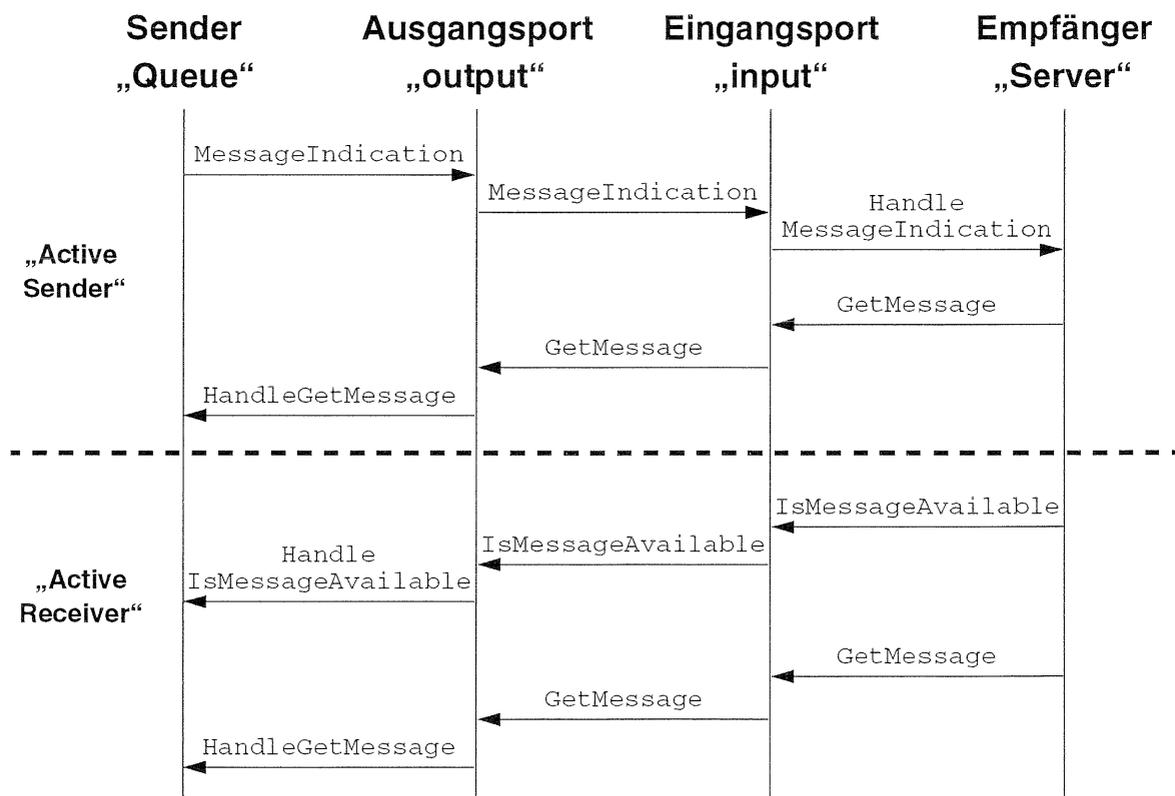


Meldungsaustausch zwischen Modellkomponenten über Ports:

- Unterscheidung von Eingangs- und Ausgangsports
- Registrierung der Ports mit der Adresse ihres Eigentümers und ihrem Namen beim zentralen Port-Manager
- Verbinden von Ports mit Funktions-Aufruf Connect
- Kommunikation mittels Handshake-Protokoll

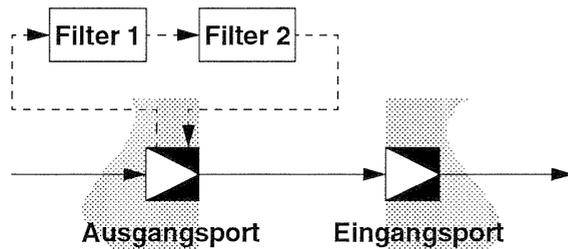
⇒ einheitliche Schnittstelle für Modellkomponenten

Kommunikations-Protokoll



Meldungsfilter

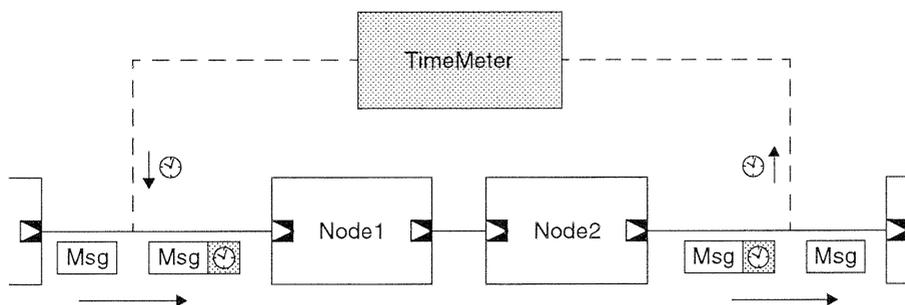
- Anbindung an Port mit Methodenaufruf
- Umleitung der Protokollfunktionen



- **Motivation:**
 - Auswertung oder Modifikation von Meldungsinhalten
 - Meldungs-/Protokoll-Tracing
 - Verwendung in Meßgeräten

Meßgeräte

- „Anklemmen“ an Ports über Meldungsfilter
- **Anwendung:**
 - Meldungsähler
 - Durchlaufzeitmessung



- **interne Erfassung der Meßwerte mit Hilfe von Statistiken**

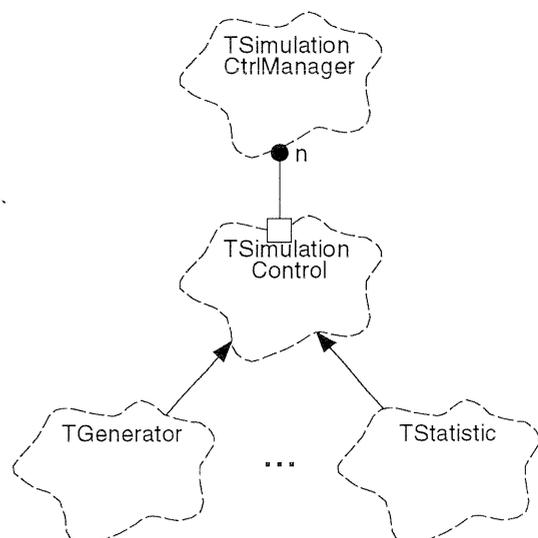
Statistische Auswertung

Eigenschaften der Statistikklassen:

- **Ermittlung von**
 - Mittelwert, Standardabweichung und Variationskoeffizient von Stichproben
 - diskretisierter Verteilungsfunktion und Verteilungsdichtefunktion
 - Mittelwert bei Zustandsprozessen
- **Berechnung von Vertrauensintervallen**
- **Verwendung in**
 - Modellkomponenten (z.B. Warteschlangenlänge)
 - Meßgeräten (z.B. Durchlaufzeit)
- **Anbindung an Ablaufsteuerung**

Ablaufsteuerung

- **Initiierung der Phasen der Simulation durch Methodenaufwurf Run beim Simulationsobjekt**
 - Initialisierung
 - Warmlaufphase
 - Teiltests
 - Ergebnisaufbereitung und -ausgabe
- **Zugang zur Ablaufsteuerung über Notifier**
- **Triggern von Objekten am Beginn und am Ende einer Phase, z.B.**
 - Generatoren
 - Statistiken



Zusammenfassung

Vor- und Nachteile des objektorientierten Ansatzes:

- ⊕ Nähe zum Problembereich
- ⊕ Strukturierung komplexer Modelle durch Hierarchie-Konzept
- ⊕ Erweiterbarkeit durch einheitliche Schnittstellen
- ⊕ Kapselung grundlegender Vorgänge

- ⊖ größere Laufzeit gegenüber prozeduraler Programmierung
- ⊖ hoher Einarbeitungsaufwand

