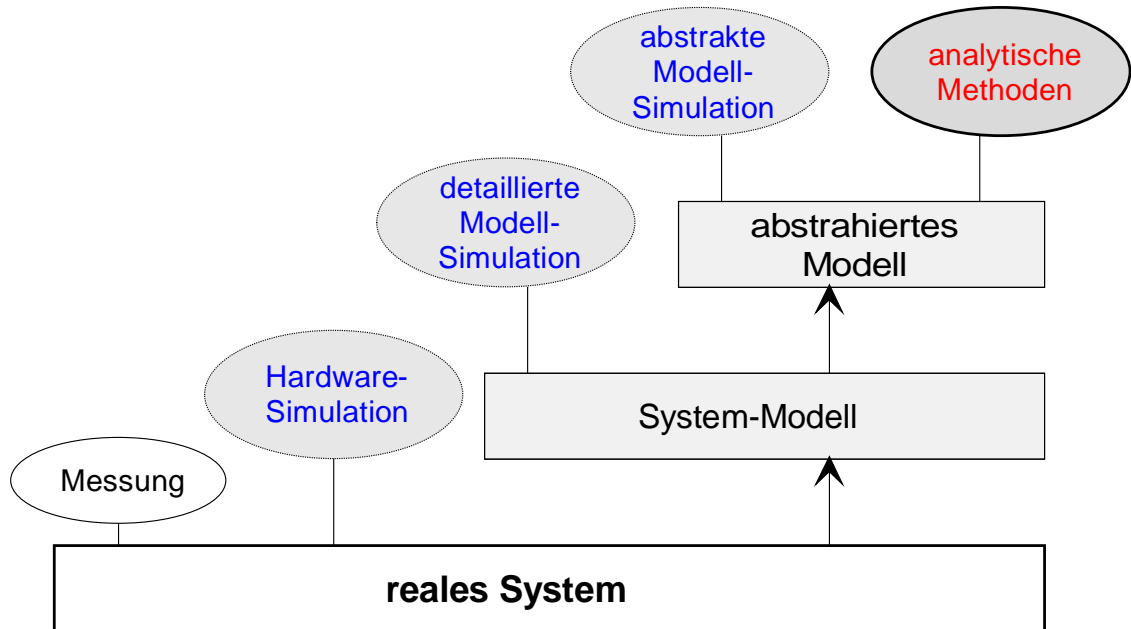


2 Einführung in die Simulationstechnik

2.1 Leistungsbewertungsmethoden und Simulation (LK 1.2)

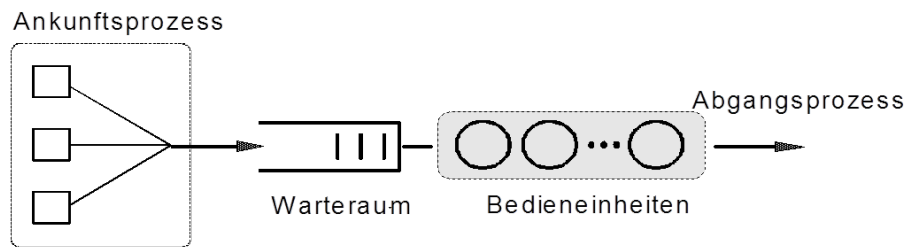


- Messung am realen System: nur möglich, wenn dieses schon existiert und zugänglich ist
- Hardware-Simulation: die Hardware eines realen Systems wird mit Hilfe von Software simuliert
- Detailliertes System-Modell: Abstraktion des realen Systems, Implementierungsdetails wie Hardware-Schaltungen spielen keine Rolle
- Detaillierte Modell-Simulation: Simulation des detaillierten System-Modells mit sehr vielen Details, die möglicherweise nicht alle einen entscheidenden Einfluss auf das Modell-Verhalten haben
- Abstrahiertes System-Modell: Abstraktion des realen Systems, nur noch die für das Verhalten des System wichtigen Eigenschaften werden z.B. als Warteschlange modelliert
- Abstrakte Modell-Simulation: Simulation des abstrahierten System-Modells
- Analytische Methoden: Direkte Quantifizierung des Verhaltens des abstrahierten System-Modells mit Hilfe von Mathematik und ohne Simulation

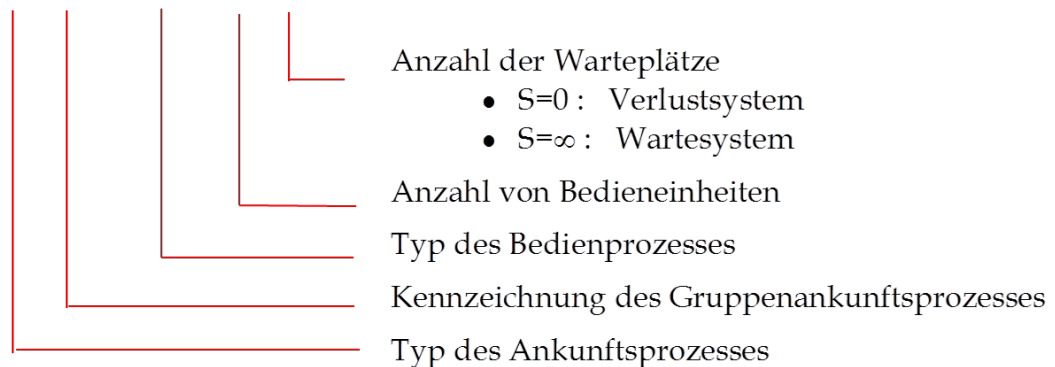
2.2 Ereignistreue Simulation und ihre Anwendungen

- Simulation: Nachbildung der Struktur und des Zustandes des Systems sowie seiner dynamischen Entwicklung in einem Programm
- Zeittreue bzw. ereignistreue Simulation: Simulationsverfahren, bei dem sich der Systemzustand nur zu bestimmten Ereignissen verändert; diese Ereignisse sind mit einem Zeitpunkt assoziiert, der eine Entsprechung im Realsystem hat.
- Hauptaufgaben:
 - Bei Untersuchung eines Systems, dessen Verhalten relativ unbekannt ist, nutzt man ereignistreue Simulation um das Verhalten des Systems zu erforschen.
 - Simulation ist notwendig, solange das System nicht mit analytischen Methoden modelliert und untersucht werden kann.
 - Simulationsergebnisse helfen analytische Modelle zu finden, denn sie geben Hinweise auf das Verhalten eines Systems
 - Simulationen dienen auch zur Validierung von Ergebnissen, die auf Basis analytischer Modelle und Methoden gewonnen wurden
 - Simulationsergebnisse zu produzieren ist bei komplexen Systemen oft einfacher, aber analytische Modelle haben den Charme, dass sie ein gewisses Systemverständnis ausdrücken.
 - Simulation wird auch zur Veranschaulichung bzw. Animation von Systemabläufen genutzt.

2.3 Notation für einstufige Warteschlangenmodelle



$GI^{[X]}/GI/n-S$ (Kendall-Notation)

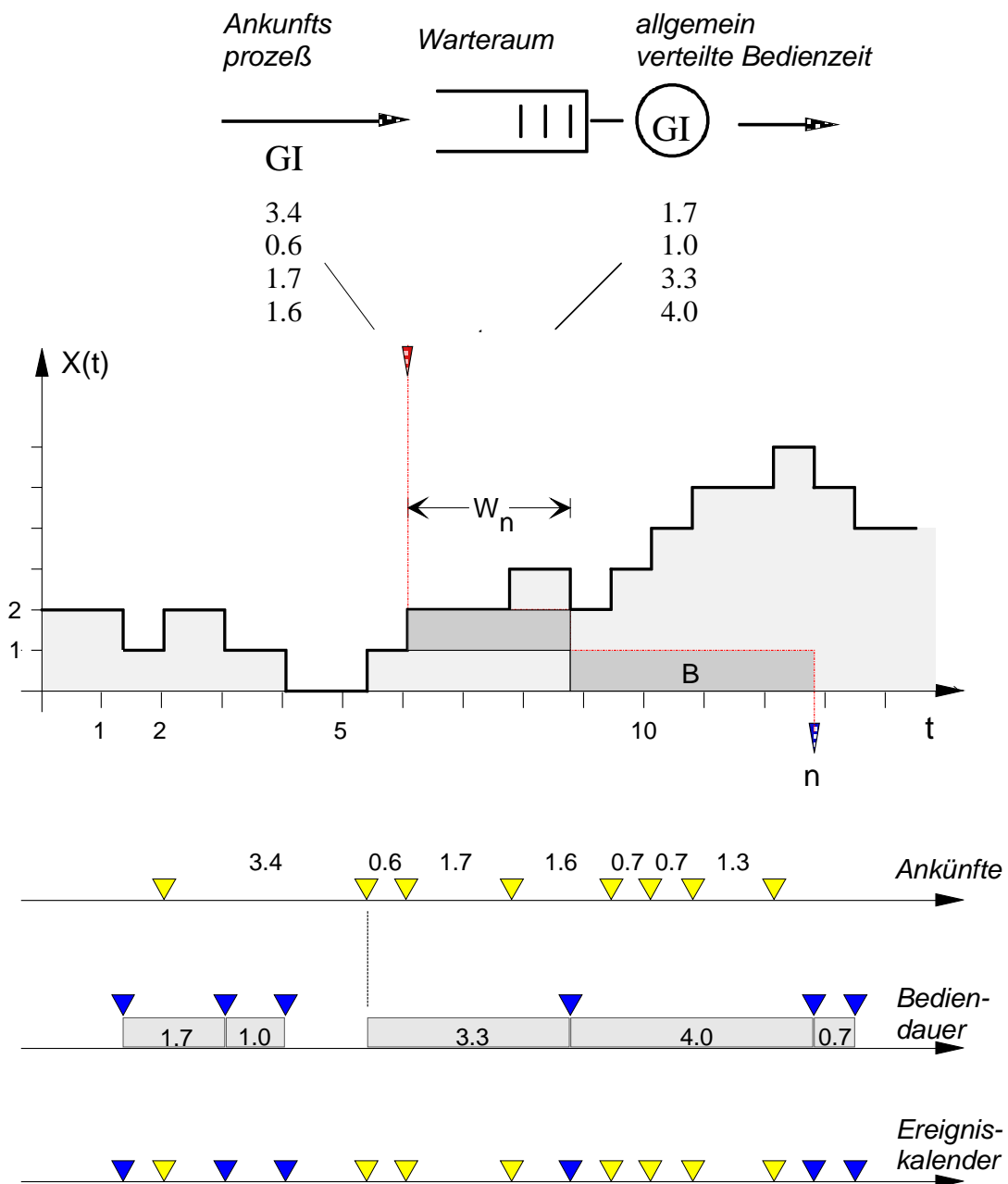


- **GI** : (general independent)
 - Ankunftsprozesse mit Erneuerungseigenschaft (Erneuerungsprozess)
 - Bedienprozesse, die jeweils mit Hilfe einer Zufallsvariablen beschrieben werden können;
 - Realisierungen dieser Zufallsvariablen sind statistisch unabhängig voneinander
- **D** : Deterministisch
- **M** : Markov
 - Die zugehörige Zufallsvariable ist exponentiell verteilt.
 - Ein M-Ankunftsprozess ist dementsprechend ein Poisson-Prozess
- **E_k** : Erlang-k-verteilt
- **H_k** : Hyperexponentiell verteilt, k-ter Ordnung

2.4 Komponenten und Organisation ereignistreuer Simulationen

2.4.1 Einführendes Beispiel: Simulation eines einstufigen GI/GI/1-S Wartesystems

- Anwendungsbeispiel: Ausgehendes Interface eines Routers
- Zeitdiskrete Simulation des theoretischen Modells
- Systemzustand $X(t)$: Anzahl der Anfragen im System
- Änderungen des Systemzustands durch Ankunfts- und Bedienendeereignisse



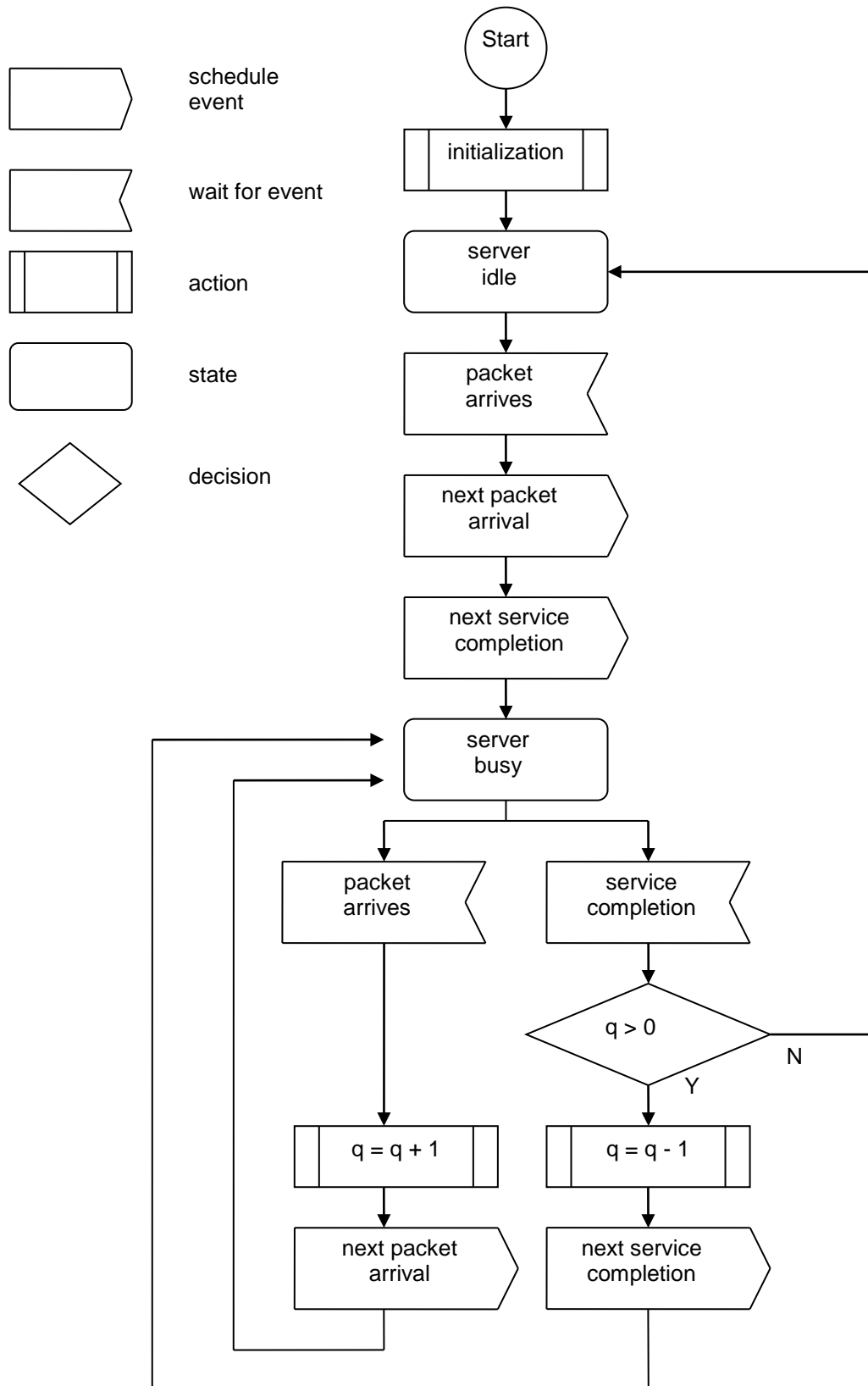
2.4.2 Aufbau einer „Discrete Event Simulation (DES)“ (LK 1.3.2)

- Simulation Clock (Simulationsuhr)
 - Wiedergabe der aktuellen Systemzeit
 - Implementierung der Simulationszeit üblicherweise als *long int* → Warum?
 - Umrechnung von Echtzeit auf Simulationszeit notwendig
- Event (Ereignis)
 - Mit einem diskreten Simulationszeitpunkt assoziiert
 - Beispiele: nächste Paketankunft, nächstes Bedienende
 - Führt zu diesem Zeitpunkt eine Event Routine aus, die eine Aktion im System verursachen, z.B. den System State verändern kann.
 - Verwaltung in der Event Chain (Ereignisliste)
- Statistic Counters
 - Variablen zur Speicherung von statistischen Informationen über das Systemverhalten
 - Discrete counters, z.B. Paketgrößen
 - Time-weighting counters, z.B. Warteschlangenlängen
 - Momente, Konfidenzintervalle, Output-Generierung
- Hauptprogramm
 - Definition und Initialisierung des System State (Systemzustand: Inhalt von Variablen, Listen, ...)
 - Initialisierung der Event Chain
 - Kontrolle und Abarbeitung der Event Chain
 - Bestimmung des nächsten Ereignisses in der Event Chain
 - Vorstellen der Simulationsuhr auf den Zeitpunkt dieses Ereignisses
 - Ausführen der Event Routine dieses Ereignisses
 - Kontrolle der Simulationsdauer
 - Anzahl bestimmter Events
 - Vorgegebene Simulationszeit

- Statistische Signifikanz
- Report Generator (Ergebnisroutine):
 - Verständliche Ausgabe der Simulationsergebnisse am Ende der Simulation
 - Leistungsmaße basierend auf den Statistic Counters
- Bibliotheksroutinen (*library routines*) zur
 - Erzeugung von Zufallsgrößen, Erhebung von Statistiken, ...

2.4.3 Hauptmodellierungsansätze (LK 3.3)

- Ereignisorientiert (*event-scheduling approach*)
 - Sequentielle Abarbeitung der Ereignisse nach obigem Konzept (aufsteigende Simulationszeit)
 - Kein Fortschreiten der Simulationszeit während der Bearbeitung eines Ereignisses, Beispiele:
 - Inkrementieren des Warteschlangenfüllstandes bei Paketankunft
 - Dekrementieren des Warteschlangenfüllstandes bei Bedienung
 - Übliches Konzept für einfache Eigenimplementierungen (siehe Übung!)
- Prozessorientiert
 - Fokus der Betrachtung: Objekt (z.B. Warteschlange)
 - Definition des dazugehörigen Prozesses durch ein Ablaufdiagramm, das auf Ereignisse reagiert \Rightarrow Ein Prozess ist eine zeitlich geordnete Folge von zusammenhängenden Ereignissen, die ein Objekt (d.h. Kunde, Job, Teil, etc.) erfährt.
 - Fortschreiten der Simulationszeit während eines Prozesses
 - Im Allgemeinen mehrere Eintrittspunkte für einen Prozess
 - Wegen Einfachheit Verwendung zur Modellbeschreibung in professionellen Simulationssprachen (z.B. OpNet)
 - Interne Umwandlung in Ereignisse
 - Beispiel: Prozessorientierte Modellbeschreibung eines GI/GI/1- ∞ Warteschlangensystems



2.5 Klassifikation von Simulationsmodellen (LK 1.2)

- Deterministisch vs. stochastisch
 - *Deterministisch*: System enthält keine zufallsabhängigen Komponenten (z.B.: Differentialgleichungen zur Beschreibung chemischer Reaktion)
 - *Stochastisch*: Systemverhalten wird durch zufällige Ereignisse beeinflusst (z.B.: Warteschlangen- oder Lagerhaltungssysteme)
- Kontinuierlich vs. diskret
 - *Kontinuierlich*: Systemzustände ändern sich kontinuierlich (z.B. chemische Reaktionen, Differentialgleichungssysteme)
 - *Diskret*: Systemzustände ändern sich an diskreten Zeitpunkten (z.B. Lagerhaltungssysteme)
- Statisch vs. dynamisch
 - *Statisch*: System nur an einem Zeitpunkt betrachtet oder Zeit spielt keine Rolle (z.B.: Monte-Carlo-Simulation, LK 1.8.3, Auswürfeln von Snapshots mit nachfolgender deterministischer Auswertung)
 - *Dynamisch*: Modell repräsentiert das zeitliche Verhalten des Systems (z.B.: Simulation von Warteschlangen)

Definition: Eine *ereignistreue Simulation* (*discrete event simulation*, kurz: DES) ist die Modellierung der zeitlichen Entwicklung eines Systems durch eine Darstellung, bei der sich die Zustandsvariablen des Systems an diskreten Zeitpunkten augenblicklich ändern. Diese Zeitpunkte werden als *Ereignisse* bezeichnet.

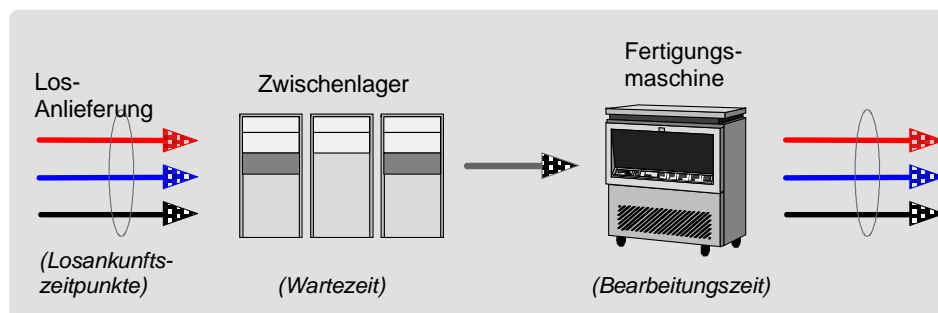
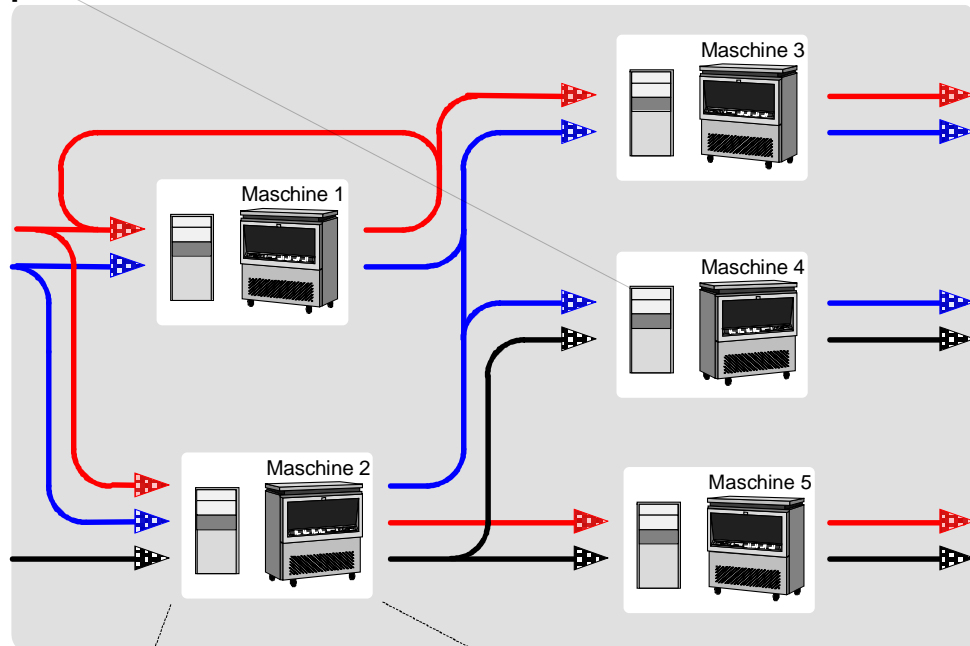
Bemerkung: Es gibt aber auch Ereignisse, die nicht zu Zustandsänderungen führen, z.B. Simulationsende, zeitlich festgelegte Erfassung von Statistiken, etc.

2.6 Fokus von Simulationen

Unterschiedliche Fragestellungen bedürfen unterschiedlicher Modelle und Simulationen

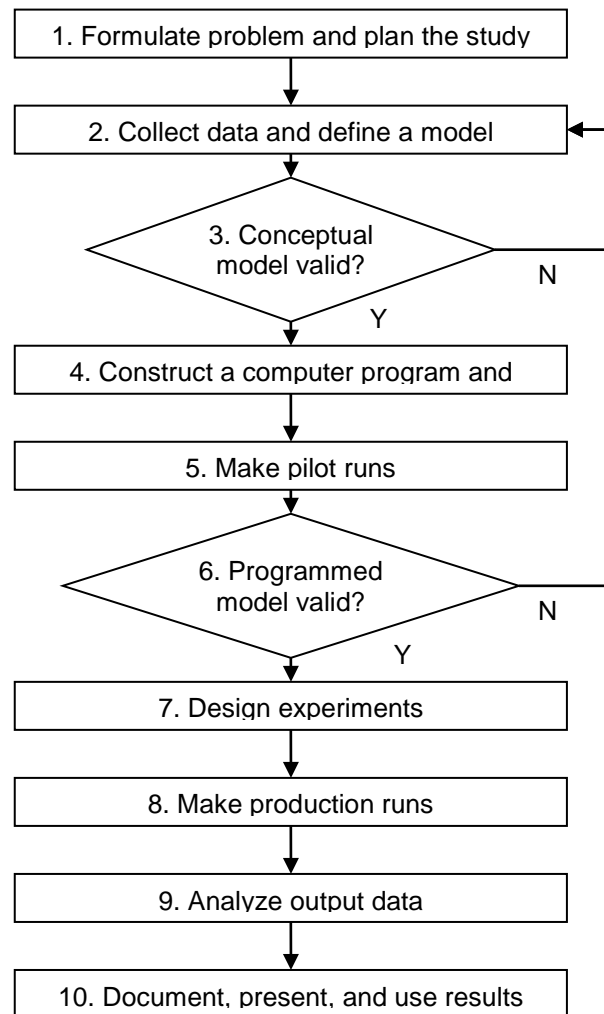
- Bestimmung der Durchlaufzeit eines Produktes in einer Fabrik bei einem bestimmten Produktmix
- Vergleich unterschiedlicher Scheduling-Mechanismen in einer Maschine

produktorientierte Simulation



maschinenorientierte Simulation

2.7 Bestandteile einer Simulationsstudie (LK 1.7)



1. Problemformulierung und Planung der Studie
 - Kickoff-Workshop, Interview mit Sachexperten
 - Festlegen der Fragestellung und der Leistungsmaße
2. Datensammlung und Modelldefinition
 - Herausfinden der wesentlichen Eigenschaften des Systems in Bezug auf die Fragestellung
 - Festlegen eines geeigneten Abstraktionsmodells
 - Erhebung statistischer Größen über das zu untersuchende System (z.B. Anzahl der Maschinen, Verteilungen von Zwischenankunftszeiten, Annahmen über nicht messbare Größen)
3. Validierung des Konzeptes
 - Rücksprache mit den Sachexperten

4. Erstellung und Verifikation eines Simulationsprogramms
 - Wahl einer Programmiersprache (C++, Java, ...) oder eines Simulationstools (ns3, OMNeT++, OpNet, ...)
 - Debugging
5. Pilotläufe
6. Validierung des Simulationsprogramms
 - Vergleich der Ergebnisse mit originalen Messdaten (falls möglich)
 - Plausibilitätsprüfung der Ergebnisse zusammen mit Sachexperten
 - Durchführung von Sensitivitätsanalysen zur Bestimmung signifikanter Einflüsse auf die Systemleistung mit anschließender Modellverfeinerung
7. Design von Experimenten
 - Erschöpfende Systemsimulation wegen Begrenztheit von Simulationsressourcen nicht möglich
 - Nur aussagekräftige Experimente interessant
 - Leistungsbewertung unterschiedlicher Systemkonfigurationen bei gleichen Systemmechanismen
 - Leistungsbewertung unterschiedlicher Systemmechanismen bei gleicher Systemkonfiguration
 - Verfahren zur Erhöhung der Zuverlässigkeit der Ergebnisse
 - Bestimmung geeigneter Aufwärmphasen
 - Bestimmung der Simulationslängen
8. Ergebnisläufe
9. Analyse der Simulationsergebnisse
 - Testen der statistischen Zuverlässigkeit der Ergebnisse (Konfidenzintervalle)
 - Bewertung konkurrierender Systementwürfe
10. Dokumentation, Präsentation und Nutzung der Ergebnisse
 - Dokumentation des Modells, der Annahmen, der Experimente, der Ergebnisse
 - Graphische Aufbereitung zur Vermittlung der Ergebnisse, Erklärung der Ergebnisse, Vertrauen in die Ergebnisse durch Diskussion des Modells und seiner Validierung
 - Nutzung der Ergebnisse in Entscheidungsprozessen (durch den Auftraggeber)

Bemerkungen

- Nicht alle Simulationsstudien enthalten notwendigerweise jeden der obigen Schritte.
- Oft mehrere Zyklen in der Praxis
 - Systemkenntnis wächst mit dem Studienfortschritt
 - Modelländerungen auch in späteren Schritten möglich