

2 Elemente zeitdiskreter Simulationsmodelle

Obwohl zeitdiskrete Simulationsmodelle in vielfältigen Formen auftreten und für die unterschiedlichsten Problemstellungen eingesetzt werden, lassen sich typische Merkmale und wesentliche Grundelemente identifizieren, die in jedem dieser Modelle auftreten. So muß in jedem Modell die Beziehung zwischen Systemzustand und Simulationszeit hergestellt werden (2.1). Diesbezüglich haben sich in der zeitdiskreten Simulation unterschiedliche konzeptuelle Sichtweisen herausgebildet, die auch als *Weltbilder* oder *Modellierungsstile* bezeichnet werden und eng mit bestimmten Simulationssprachen verbunden sind. Dazu gehören im wesentlichen der **ereignisorientierte**, der **prozeßorientierte**, der **transaktionsorientierte** und der **aktivitätsorientierte Ansatz** der Simulation (2.2), wobei der letztgenannte allerdings wenig praktische Bedeutung erlangt hat. Im Mittelpunkt der Darstellung steht der ereignisorientierte Ansatz, dessen prinzipielle Modellkomponenten und Abläufe in 2.3 ausführlich behandelt werden. In 2.4 werden typische Problemklassen dargestellt, die in der diskreten Simulation untersucht werden. Hierzu gehören Bedienungs-/Wartesysteme (2.4.1), Lagerhaltungssysteme (2.4.2) und ausfallanfällige Systeme (2.4.3). Für jede dieser Problemklassen, denen bestimmte Modellklassen zugeordnet werden können, wird ein überschaubares Anwendungsbeispiel präsentiert. Gegenstand von 2.5 ist schließlich die Gegenüberstellung der beiden wichtigsten Modellierungsstile der zeitdiskreten Simulation, des ereignisorientierten und des prozeßorientierten Ansatzes, anhand eines konkreten Beispiels, dem Modell eines Fertigungssystems.

2.1 Beziehungen zwischen Zustand und Zeit

Bei der Modellierung eines Systems muß nicht nur ein bestimmter Zustand abgebildet werden, es ist vielmehr darzustellen, wie sich der Systemzustand in Abhängigkeit von der Zeit verändert. In diesem Abschnitt werden die grundlegenden Konzepte eingeführt, die in der zeitdiskreten Simulation die Beziehung zwischen dem Systemzustand und der (Simulations-) Zeit herstellen. Es geht also um die prinzipiellen Möglichkeiten, den Zusammenhang zwischen der (statischen) Struktur und dem (dynamischen) Verhalten eines Systems in einem diskreten Modell abzubilden.

Die statische Struktur eines Systems kann man sich aus einer Vielzahl von Objekten bestehend vorstellen, die durch ein Netz von Beziehungen miteinander verbunden sind. Die Objekte werden durch eine Anzahl von Attributen charakterisiert, die für jedes individuelle Objekt bestimmte (quantitative oder qualitative) Werte annehmen. Im weitesten Sinne können auch die Beziehungen als Attribute aufgefaßt werden. Somit sind zwei Arten von Attributen zu unterscheiden:

- *Indikative Attribute*: Sie beschreiben die Eigenschaften von Objekten (Beispiel: Ein Auftrag hat eine bestimmte Bearbeitungsdauer).
- *Relationale Attribute*: Sie setzen Objekte zueinander in Beziehung (Beispiel: Zur Bearbeitung bestimmter Aufträge werden bestimmte Maschinen benötigt).

Der *Zustand eines Objektes* sei nun das n-Tupel aller Attributwerte dieses Objektes. Der Zustand des gesamten Systems sei die Menge aller Objektzustände. Um das dynamische Verhalten eines Systems abzubilden, müssen Attributwerte im Modell mit einem Zeitindex versehen werden. Der Index kann als spezielles Attribut, als *Indexattribut* aufgefaßt werden, das allen Objekten gemeinsam ist. Neben der **Zeit** sind auch andere Indexattribute denkbar, etwa wenn die **räumliche** Dynamik von Systemen durch Simulation untersucht werden soll. Beispiele hierfür sind die Modelle der Schadstoffausbreitung, die im Umweltschutz eingesetzt werden.

Das Indexattribut Zeit wird auch als *Zeitbasis* eines Simulationsmodells bezeichnet. Die **diskrete**, abzählbare Zeitbasis, die in zeitdiskreten Simulationsmodellen verwendet wird, bedeutet eine Einschränkung: Vorgänge, deren exakte Beschreibung eine **kontinuierliche** Zeitbasis erfordern würde, werden von der Modellierung ausgeschlossen. In jedem Fall besteht aber die Möglichkeit einer Approximation kontinuierlicher Vorgänge durch zeitdiskrete Modelle. Bei der Verwendung eines Digitalrechners als "Modellmedium" ist die Diskretisierung kontinuierlicher Vorgänge sogar prinzipiell unumgänglich. Beruht das konzeptuelle Modell auf der Vorstellung eines kontinuierlichen Ablaufs und wird dieser nur aus technischen Gründen diskretisiert, so spricht man auch von einem *quasikontinuierlichen* Modell.

Die Beziehung zwischen Zustand und Zeit kann in einem diskreten Simulationsmodell nun auf dreierlei Weise hergestellt werden:

- durch ein **Ereignis** (*event*), das die Veränderung eines Objektzustandes zu einem bestimmten Zeitpunkt, dem *Ereigniszeitpunkt*, bewirkt. Dabei sind zwei Arten von Ereignissen zu unterscheiden:
 - *exogene Ereignisse*: Der Ereigniszeitpunkt ist von der Systemumgebung vorgegeben; es besteht keine interne Abhängigkeit von anderen Ereignissen (Beispiel: Eintreffen eines Kunden)
 - *endogene Ereignisse*: Sie treten als Folge von Zustandsveränderungen ein, wobei die Ereigniszeitpunkte determiniert oder stochastisch sein können (Beispiel: Ende der Bedienung eines Kunden)

- durch eine **Aktivität** (*activity*), die aus einer Menge von Operationen besteht, die während eines Zeitintervalls ausgeführt werden. Die Wirkung einer Aktivität, also die Zustandsänderung des Objekts, wird dem Endzeitpunkt des Intervalls zugeordnet.
- durch einen **Prozeß** (*process*), der aus einer Folge von Aktivitäten besteht, die auf ein bestimmtes Objekt bezogen sind und während einer Zeitspanne ablaufen.

Wie Abbildung 2-1 veranschaulicht, können Systemabläufe auf elementarer Ebene stets durch Ereignisfolgen gekennzeichnet werden. Bei der **diskreten Ereignis-Simulation** (*discrete event simulation*) wird eine modellinterne Simulationsuhr von der Ablaufkontrolle jeweils bis zum nächstfolgenden Ereigniszeitpunkt vorgerückt (s. 2.3). Dies impliziert, daß die Zeit zwischen zwei benachbarten Ereignissen im allgemeinen variabel ist. Der Simulationszeitpunkt eines Ereignisses kann *beliebig* gewählt werden, wodurch eine wirklichkeitstreue Abbildung der Zustandsänderungen möglich ist. Dies wäre nicht der Fall, wenn man die Simulationsuhr in konstanten Zeitintervallen weiterzuschalten würde. Dann müßten alle Ereigniszeitpunkte auf das Ende eines solchen Intervalls verschoben werden, was bei ungünstiger Wahl der Schrittlänge erhebliche Verfälschungen des Modellverhaltens bewirken kann.

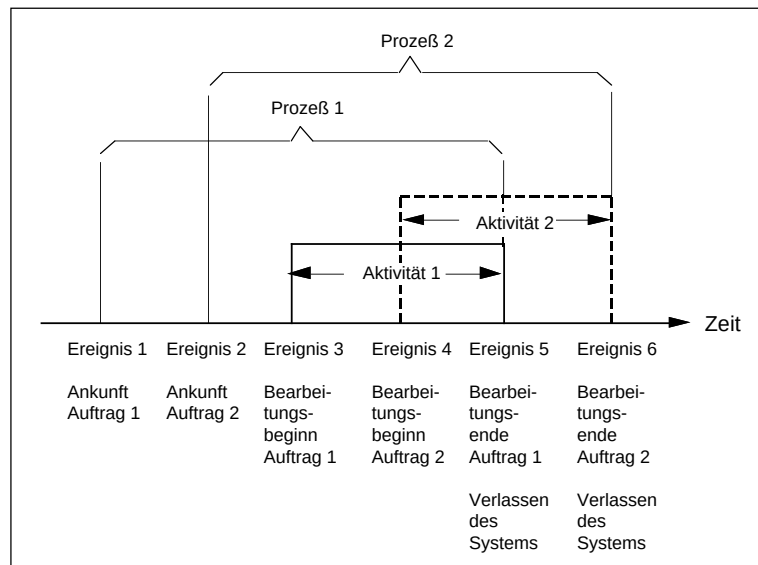


Abb. 2-1: Zusammenhang von Ereignis, Aktivität und Prozeß am Beispiel der Auftragsbearbeitung

Nun ist es durchaus möglich, daß Ereignisse simultan auftreten, also in einem Ereigniszeitpunkt zusammenfallen. Ein sequentiell arbeitender Rechner ist jedoch nur in der Lage, die Ereignisse (Zustandsänderungen) nacheinander abzuarbeiten: Das erzwingt eine Sequentialisierung paralleler Ereignisse. Hierfür wird von Simulationssprachen (und anderen Klassen von Simulationssoftware, s. 6.1) in der Regel ein **Prioritätsmechanismus** bereitgestellt.

2.2 Konzepte der zeitdiskreten Simulation

Innerhalb der zeitdiskreten Simulation haben sich verschiedene konzeptuelle Sichtweisen herausgebildet, auch **Weltbilder** (*world views*) oder **Modellierungsstile** (*modelling styles*) genannt. Sie wurden jeweils durch paradigmatische Simulationssprachen geprägt. Mit der Verwendung einer bestimmten Simulationssprache ist somit eine bestimmte Sichtweise der Realsysteme und ein Modellierungsstil verbunden.

Die vier klassischen Weltbilder (s. Abb. 2-2) unterscheiden sich in erster Linie in ihrer Sichtweise und Darstellung der Dynamik des Systemverhaltens, also den oben erwähnten Beziehungen zwischen Zeitablauf und Zustandsveränderungen eines Systems. Das hat zur Folge, daß der eine oder andere Modellierungsstil je nach Art des zu modellierenden Realsystems natürlicher erscheint.

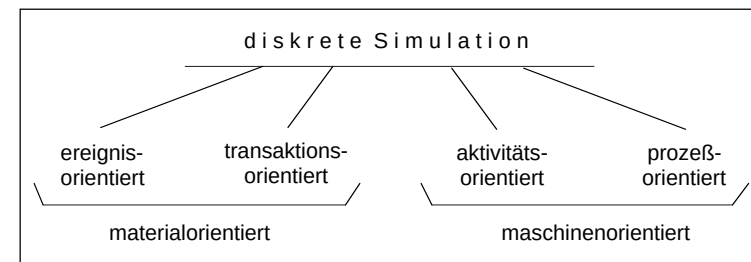


Abb. 2-2: Die "Weltbilder" der diskreten Simulation

Während in den USA die sogenannten materialorientierten Konzepte dominieren (ereignis- und transaktionsorientierter Ansatz), ist in Europa eher das maschinenorientierte Denken vorherrschend (prozeß- und aktivitätsorientierter Ansatz). *Materialorientiert* bedeutet in diesem Zusammenhang, daß bei der Modellbildung primär die Materialflüsse durch Bedienstationen und die dabei zurückzulegenden Wege betrachtet

werden. (Der Begriff der Bedienstation wird hier wesentlich weiter gefaßt als in der Umgangssprache; vgl. 2.4.1.) Ein materialorientiertes Modell des Kundenflusses durch einen Supermarkt würde z. B. das Aufsuchen bestimmter Verkaufsstände als Bedienstationen aus der Perspektive der Kunden darstellen. Bei der *maschinenorientierten* Sicht steht dagegen der Bearbeitungsvorgang selbst im Vordergrund. Hier würde also primär das Geschehen an den einzelnen Verkaufsständen modelliert.

In einigen Darstellungen wird das transaktionsorientierte Weltbild dem prozeßorientierten implizit untergeordnet (vgl. z. B. [Kre86]). Die hier vollzogene Trennung ist jedoch insoweit einsichtig, als sich die Implementationssprache, die diesen Modellierungsstil geprägt hat, die Simulationssprache **GPSS**, grundlegend von prozeßorientierten Sprachen wie **Simula** unterscheidet. Als ein Vertreter des ereignisorientierten Weltbildes sei hier **Simscrip** genannt, und **ECSL** möge für die aktivitätsorientierten Sprachen stehen (s. dazu [Kre86] und Abschnitt 6.1).

Der aktivitätsorientierte Ansatz hat nicht die Bedeutung der anderen Ansätze erlangt und soll deshalb nicht weiter vertieft werden. Nur soviel sei erwähnt: Im Gegensatz zum ereignisorientierten Ansatz, bei dem Ereignisroutinen durch das Abarbeiten einer Ereignisliste aktiviert werden, werden hier in einem eher deklarativen Stil Bedingungen für den Beginn und das Ende von Aktivitäten spezifiziert. Diese Bedingungen, die beliebig komplex sein können, müssen für alle Aktivitäten zu jedem Simulationszeitpunkt überprüft werden, was erhebliche Effizienzprobleme aufwirft.

Die Grundzüge der anderen drei Simulationsansätze werden in den folgenden Abschnitten vorgestellt. Zur Vertiefung der beiden wichtigsten Konzepte, der ereignisorientierten und der prozeßorientierten Simulation, dienen die anschließenden Beispiele. Die Diskussion der unterschiedlichen Simulationskonzepte (einschließlich des aktivitätsorientierten Ansatzes) wird in 7.2 und 7.12 unter Implementationsaspekten nochmals aufgegriffen.

2.2.1 Ereignisorientierte Sicht (event scheduling)

Dieser Ansatz beruht auf einer detaillierten Beschreibung von Ereignissen in Form von *Ereignisroutinen* (*event routines*). In einem ereignisorientierten Modell werden lediglich die **Zustandsänderungen** nachvollzogen, die zu den Ereigniszeitpunkten stattfinden, nicht jedoch die Tätigkeiten (Aktivitäten), die in den dazwischenliegenden Intervallen ablaufen. Zeitlich ausgedehnte Vorgänge werden somit auf eine Folge von Ereignissen reduziert, von denen jedes einzelne einem Punkt auf der Zeitachse (dem Ereigniszeitpunkt) zugeordnet ist. Unabhängig davon, wieviel Rechenzeit zur Ausführung einer Ereignisroutine auf einem Computer real verbraucht wird, geschieht sie konzeptuell **zeitverzugslos**. Die interne Simulationsuhr wird vor der Ausführung der Ereignisroutine zum Ereigniszeitpunkt vorgestellt, steht während der Ausführung still

und springt nach vollzogener Zustandsänderung auf den Zeitpunkt des nächsten Ereignisses vor.

Die ereignisorientierte Sicht erlaubt eine klare Trennung zwischen der Struktur und dem Verhalten des zu simulierenden Systems. Im Modell werden unterschieden:

- *Statische Komponenten:* Die Datenstrukturen, die die unterschiedlichen Objektklassen des Realsystems und die permanent vorhandenen individuellen Objekte (Ausprägungen dieser Klassen, z. B. Maschinen eines bestimmten Maschinentyps) abbilden. Die vorgesehenen Attribute ermöglichen die Kennzeichnung der individuellen Objekte und legen die prinzipiell möglichen Zustandsänderungen fest.
- *Dynamische Komponenten:* Die im Modell vorgesehenen Ereignisse (z. B. Beginn der Bearbeitung eines Werkstücks), dargestellt durch sogenannte Ereignisroutinen, und temporäre Objekte, die im Laufe der Simulation durch Ereignisroutinen erzeugt oder vernichtet werden (z. B. Werkstücke).

Die **Ereignisroutinen** bestimmen in ihrem Zusammenspiel das Verhalten des Modells, indem sie

- Attributwerte von Objekten ändern,
- temporäre Objekte generieren oder löschen,
- einer Ereignisliste neue, geplante Ereignisse hinzufügen oder aus der Liste streichen.

Die **Ablaufkontrolle** erfolgt durch einen Programmteil, der die in einer Ereignisliste nach ihren Ereigniszeitpunkten geordneten Ereignisse sequentiell abarbeitet (*next event approach*). Als wesentlich ist nochmals hervorzuheben, daß bei diesem Ansatz die sogenannte *tote* Zeit, also die Zeit zwischen den Ereigniszeitpunkten, übersprungen wird.

Die ereignisorientierte Formulierung eines Modells liegt nahe bei Vorgängen, die im wesentlichen im **Belegen und Freigeben von Bedienstationen** bestehen, ohne daß die Modellkomponenten komplex interagieren. Die Ablaufsteuerung ist wesentlich einfacher zu realisieren und die Programme laufen meist auch schneller als prozeß-, transaktions- oder aktivitätsorientierte Modelle.

2.2.2 Prozeßorientierte Sicht (process interaction)

Im prozeßorientierten Weltbild werden die auf ein Objekt bezogenen Aktivitäten mit den Objektattributen in ihrer Gesamtheit zu einem **Prozeß** zusammengefaßt. Sowohl die aktiven als auch die passiven Phasen einer Systemkomponente können relativ wirklichkeitsgetreu abgebildet werden.

Während seiner aktiven Phasen führt ein Prozeß Zustandsänderungen durch, die konzeptuell zeitverzugslos ablaufen, d. h. die Simulationsuhr steht während der aktiven Phasen still. Das dabei ausgeführte Teilstück der Prozeßroutine ist in dieser Hinsicht

mit einer Ereignisroutine vergleichbar. Ein Prozeß hat nach Durchführung der Zustandsänderungen – im Gegensatz zu einer Ereignisroutine, die dann stets terminiert – jedoch die Möglichkeit, in einen inaktiven Zustand einzutreten. Die Ausführung des Prozesses kann dann zu einem späteren Zeitpunkt fortgesetzt, d. h. der Prozeß kann *reaktiviert* werden. Nach der Reaktivierung wird das nächste Teilstück der Prozeßroutine abgearbeitet.

Weil in aktiven Prozeßphasen keine Simulationszeit verbraucht wird, dienen sie zur Darstellung von Ereignissen. Die (zeitkonsumierenden) Aktivitäten einer Systemkomponente und ihre passiven Phasen werden dagegen durch *inaktive* Prozeßzustände abgebildet. Um z. B. die Bearbeitung eines Werkstücks durch eine Maschine während eines Zeitintervalls Δt darzustellen, wird man einen Maschinenprozeß definieren, der seine Reaktivierung nach Ablauf von Δt Zeiteinheiten vormerkt und sich gleichzeitig deaktiviert. Diese Operation entspricht dem Ereignis "Beginn der Bearbeitung". Die Ablaufsteuerung sorgt dann für die Reaktivierung des Prozesses zum vorgemerkten Zeitpunkt, sofern kein anderer Prozeß diesen verschoben oder gelöscht hat. Dies entspricht dem Ereignis "Ende der Bearbeitung". Die Phase, während der der Prozeß (programntechnisch) nicht aktiv war, entspricht (inhaltlich) der Aktivität "Bearbeiten".

Ein Prozeß muß den Zeitpunkt seiner Reaktivierung nicht notwendigerweise selbst bestimmen. Der häufigere Fall besteht vielmehr darin, daß ein Prozeß sich selbst passiviert und später durch andere Prozesse zur Reaktivierung vorgemerkt wird.

Ein Prozeß kann während seiner aktiven Phase

- Objektattribute modifizieren,
- neue Prozesse generieren,
- die Aktivierung anderer Prozesse zu bestimmten Zeitpunkten planen oder geplante Aktivierungen verschieben oder löschen,
- sich selbst deaktivieren, wobei die Kontrolle an einen anderen Prozeß übergeht,
- Prozesse beenden.

Die **Ablaufkontrolle** sorgt dafür, daß Prozesse in der richtigen Reihenfolge und in den vorgesehenen Zeitabständen ablaufen. Faßt man die (Re-) Aktivierung eines Prozesses als Ereignis auf, so läßt sich die Ablaufsteuerung ähnlich wie im ereignisorientierten Weltbild beschreiben: Die zu aktivierenden Prozesse werden in einer nach Zeitpunkten geordneten Ereignisliste notiert. Eine Hauptroutine – die in diesem Konzept selbst wieder als Prozeß betrachtet wird – übernimmt die Abarbeitung dieser Liste, d. h. sie wählt das jeweils nächste (Aktivierungs-) Ereignis aus und stößt den betreffenden Prozeß an. Im Gegensatz zu einer Ereignisroutine wird der Prozeß jedoch nicht notwendigerweise von seinem Anfang an abgearbeitet. Falls er bereits einmal aktiv war, wird seine Abarbeitung an der Stelle fortgesetzt, an der er sich selbst deaktiviert hat. Es ist also notwendig, den Status eines Prozesses zum Zeitpunkt seiner Deaktivierung aufzubewahren. Dazu gehören sowohl die Werte seiner Attribute als auch die Information, an welcher Stelle er fortzusetzen ist.

Im prozeßorientierten Ansatz entfällt die Auflösung logisch zusammengehöriger Abläufe auf verschiedene Ereignisroutinen, die mit dem Nachteil einer gewissen Unübersichtlichkeit der Systemstruktur verbunden ist. Bei der Modellierung **paralleler Vorgänge mit komplexen Interaktionen** bietet der prozeßorientierte Ansatz daher erhebliche Vorteile gegenüber dem ereignisorientierten.

2.2.3 Transaktionsorientierte Sicht (transaction flow)

Grundlage dieses Konzepts ist eine aus der Systemanalyse bekannte Blockdiagrammtechnik zur Beschreibung des Systemverhaltens. Die wesentlichen Elemente sind neben den *Blöcken (blocks)* mit fest vorgegebenen Funktionen die *Transaktionen (transactions)*, die auf ihrem Weg durch die einzelnen Blöcke verändert werden.

Die Blöcke sind permanent vorhanden, sie bilden also die **statischen Systemkomponenten**. Dazu gehören z. B.

- *Bedienstationen (facilities)* unterschiedlichster Art, z. B. Maschinen oder Maschinenkomplexe,
- *Speicher (storages)*, die Lager mit vorgegebener Kapazität darstellen,
- *Leitstellen (logic switches)*, die einzelne Transaktionen gemäß bestimmter Abhängigkeiten steuern.

Demgegenüber stellen die Transaktionen die temporären Elemente, also die **dynamischen Systemkomponenten** dar. Sie werden durch eine Anzahl von Parametern (indikativen Attributen) charakterisiert, die auf dem Weg durch das modellierte System in den einzelnen Blöcken verändert werden können. Somit wird eine Zustandsänderung durch eine Transaktion in einem Block ausgelöst. Sie kann darin bestehen, daß Bearbeitungsstationen belegt oder freigegeben werden, sich Warteschlangen vergrößern oder Speichereinhalte verändern.

Die **Ablaufkontrolle** läßt sich wiederum am besten durch Rückgriff auf den ereignisorientierten Ansatz verdeutlichen: Das Aufeinandertreffen von Transaktion und Block kann als Ereignis betrachtet werden. Die Transaktionen sind in der Ereignisliste nach dem Zeitpunkt ihres Eintritts in den nächsten Block und nach Prioritäten geordnet.

2.3 Komponenten ereignisorientierter Simulationsmodelle

Obwohl die zeitdiskrete Simulation für die Analyse einer breiten Vielfalt von Realsystemen eingesetzt wird, lassen sich gemeinsame Modellkomponenten identifizieren. Aus didaktischen Gründen wird hier vorerst nur der ereignisorientierte Ansatz betrachtet. Er kann als Grundlage für das Verständnis der anderen Ansätze dienen.

Typischerweise sind in ereignisorientierten Simulationsmodellen die folgenden Komponenten vorzufinden (vgl. 2.2.1):

- **Zustandsvariablen:** die Menge aller Variablen zur Beschreibung des Systemzustandes zu einem bestimmten Zeitpunkt
- **Simulationsuhr:** eine Variable, die den aktuellen Stand der Simulationszeit angibt
- **Ereignisliste:** eine Liste mit Zeitpunkt und Typ der geplanten Ereignisse, nach Ereigniszeitpunkten aufsteigend geordnet
- **Statistische Zähler:** Variablen zur Sammlung der statistischen Ergebnisse des Simulationslaufs
- **Initialisierungsroutine:** eine Prozedur zur Initialisierung des Simulationsmodells zum Zeitpunkt 0
- **Zeitführungsroutine:** eine Prozedur zur Selektion des nächsten Ereignisses aus der Ereignisliste und Vorstellen der Simulationsuhr auf den nächsten Ereigniszeitpunkt
- **Ereignisroutinen:** Prozeduren zur Aktualisierung des Modellzustandes in Abhängigkeit vom jeweiligen Ereignistyp (für jeden Ereignistyp existiert eine Ereignisroutine)
- **Ergebnisroutine (Reportgenerator):** eine Prozedur zur Berechnung der statistischen Schätzwerte der Ergebnisvariablen (anhand der statistischen Zähler) und zur Ausgabe des Ergebnisprotokolls am Ende des Simulationslaufs
- **Steuerprogramm:** ein Programmteil, der wiederholt die Zeitführungsroutine für die Bestimmung des nächsten Ereignistyps aufruft und die zugehörige Ereignisroutine aktiviert, bis der Simulationslauf beendet ist

Abbildung 2-3 zeigt das **prinzipielle Ablaufschema** der zeitdiskreten, ereignisorientierten Simulation. Die Simulation beginnt zum Zeitpunkt 0 mit dem Aufruf der Initialisierungsroutine durch das Steuerprogramm (Hauptprogramm).

Die **Initialisierungsroutine** setzt die Simulationsuhr auf 0, belegt alle Objektattribute entsprechend dem Anfangszustand des Systems, erzeugt ein oder mehrere Startereignisse, die in die Ereignisliste eingetragen werden und besetzt die statistischen Zähler mit Anfangswerten. Neben den Startereignissen wird i. d. R. ein Ereignis vom Typ *Simulationsende* erzeugt und mit der vorgegebenen Simulationsdauer als zugehörigem Ereigniszeitpunkt vormerkt.

Nach Rückkehr in das Steuerprogramm wird von dort die **Zeitführungsroutine** aktiviert. Diese bestimmt durch Zugriff auf die Ereignisliste (Entnahme des vordersten

Ereignisses) das jeweils nächstfolgende Ereignis mit Ereignistyp *i* und Ereigniszeitpunkt *t* und stellt die Simulationsuhr auf *t* vor.

Die Kontrolle wird nunmehr an das Steuerprogramm zurückgegeben, das die zugehörige **Ereignisroutine** *i* aufruft. In einer Ereignisroutine finden typischerweise die folgenden Aktionen statt:

1. Aktualisierung des Systemzustandes gemäß den Erfordernissen des Ereignistyps *i*.
2. Sammeln von Informationen (Aktualisierung der statistischen Zähler).
3. Erzeugen neuer Ereignisse, Festsetzung ihres Ereigniszeitpunktes und Eintragen in die Ereignisliste, evtl. Löschen von Ereignissen aus der Ereignisliste.

Das Eintragen eines neuen Ereignisses in die Ereignisliste (Punkt 3) wird auch als *Vormerken* oder *Ansetzen* des Ereignisses bezeichnet. Auf Programmebene wird hierfür eine sogenannte *Ereignisnotiz* erzeugt, die den Typ und den Zeitpunkt des Ereignisses verzeichnet. Zusätzlich können die Modellobjekte spezifiziert sein, deren Zustand durch das Ereignis verändert werden soll. Die Ereignisnotiz wird in die Ereignisliste eingetragen. Sie repräsentiert ein *geplantes* Ereignis, das erst dann aktuell stattfindet, wenn die Ereignisnotiz an die vorderste Stelle der Liste gelangt ist und die zugehörige Ereignisroutine aktiviert wird. In der üblichen Sprechweise wird jedoch nicht zwischen geplanten und aktuellen Ereignissen differenziert.

Wenn eine Ereignisroutine ein neues Ereignis erzeugt und vormerkt, so sind zwei Fälle zu unterscheiden. *Erstens* kann es sich um das nächste Ereignis des gleichen Typs handeln, ohne daß die beiden Ereignisse auf der konzeptuellen Ebene in einer Kausalbeziehung stehen. Dies gilt typischerweise für eine Folge von exogenen Ankunftsereignissen (beispielsweise die Ankunft von Kunden oder Aufträgen im betrachteten System). Obwohl diese Ereignisse konzeptuell voneinander unabhängig sind, ist es praktisch, ihre Abfolge im Programm so darzustellen, daß jedes Ereignis dieses Typs seinen Nachfolger erzeugt und vormerkt. Häufig sind nämlich statistische Kennwerte der Zwischenankunftszeiten solcher Ereignisse bekannt. *Zweitens* kann es sich um ein Ereignis handeln, das auf der konzeptuellen Ebene vom aktuellen Ereignis kausal abhängig ist. Beispielsweise kann der Beginn der Bedienung eines Kunden als kausale Folge seines Eintretens in einen Laden betrachtet werden. Das Vormerken eines neuen Ereignisses als Operation auf Programmebene besitzt also zwei unterschiedliche Interpretationen auf der konzeptuellen Ebene.

Nach Ausführung der Ereignisroutine erfolgt im Steuerprogramm gemäß einer vorgegebenen Abbruchbedingung die Prüfung, ob die Simulation beendet werden soll. Die Abbruchbedingung kann sich sowohl auf ein *Ereignis* vom – hierfür speziell vorgesehenen – Typ *Simulationsende* als auch auf eine vorgegebene *Simulationsdauer* beziehen. Ist die Bedingung erfüllt, also das Ende der Simulation erreicht, so wird die **Ergebnisroutine** aufgerufen, die die Abschlußarbeiten (Berechnung der statistischen Schätzwerte der Ergebnisvariablen, Reportgenerierung) durchführt. Anderenfalls wird der Ablauf mit dem erneuten Aufruf der Zeitführungsroutine fortgesetzt.

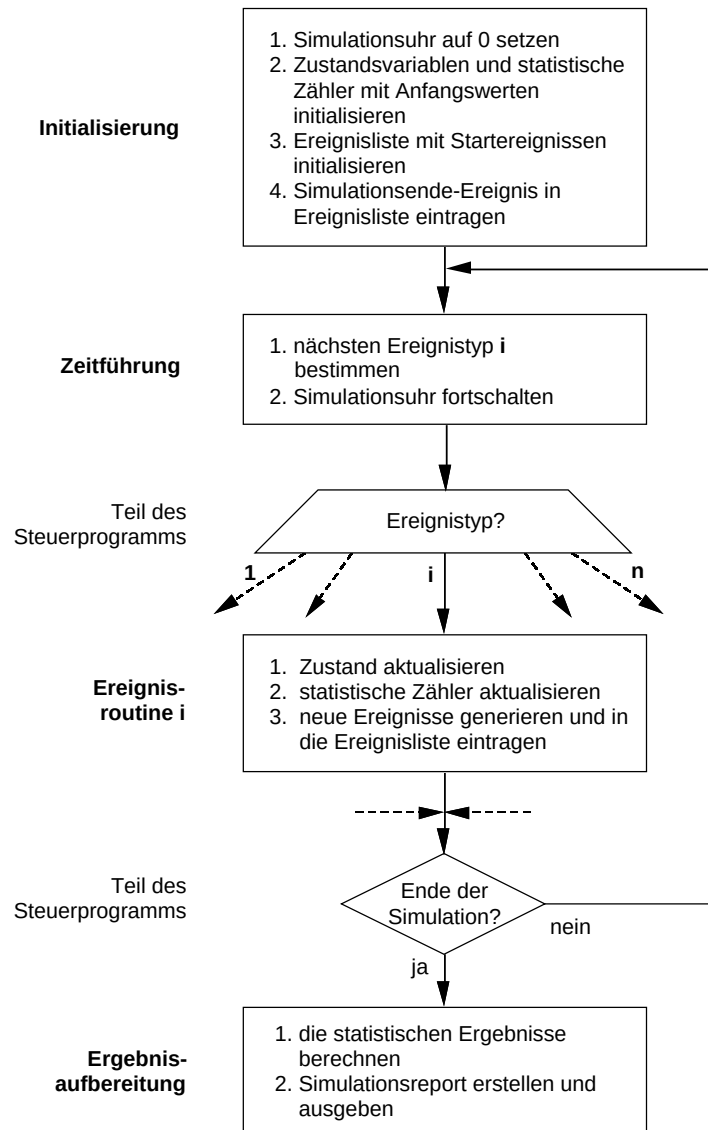


Abb. 2-3: Prinzipielles Ablaufschema der ereignisorientierten Simulation

2.4 Typische Modellklassen und ihre Realisierung

In der zeitdiskreten Simulation haben sich bestimmte Modellklassen herausgebildet, die an typischen **Problemstellungen** orientiert sind. Dazu gehört die Untersuchung des Warteverhaltens von Aufträgen oder Kunden in einem System von Bearbeitungs- oder Bedienstationen, die Abschätzung von Lagerhaltungskosten bei unterschiedlichen Bestellstrategien und die Analyse der Einsatzbereitschaft eines Systems, das wegen auftretender Fehler oder zu Wartungszwecken außer Betrieb genommen wird. Daraus ergibt sich eine Klassifikation in die drei Problembereiche *Bedienungs-/Wartesysteme*, *Lagerhaltungssysteme* und *ausfallanfällige Systeme*.

Davon zu unterscheiden sind die **Anwendungsbereiche** von Modellen. Typische Anwendungsbereiche sind z. B. Fertigungssysteme, Verkehrsplanung, Datenverarbeitung oder Gesundheitswesen, wie sie bereits in 1.5 genannt wurden. Innerhalb eines Anwendungsbereiches können verschiedene Problemstellungen untersucht und entsprechend unterschiedliche Modelle erstellt werden. So können beispielsweise an einem Fertigungssystem die Wartezeiten von Zwischenprodukten vor Maschinengruppen interessieren oder aber die Ausfallzeiten bzw. die Verfügbarkeit der Maschinen. Dementsprechend würde das Fertigungssystem als Bedienungs-/Wartesystem betrachtet (wobei die Maschinengruppen die Bedienstationen bilden) oder als ausfallanfälliges System. Auch eine Kombination beider Aspekte ist denkbar. Je nach Problemstellung kann also das gleiche Realsystem unter einem anderen Aspekt betrachtet und entsprechend in ein Modell einer anderen Klasse abgebildet werden.

In den folgenden Abschnitten werden die mit den typischen Problemstellungen verbundenen Modellklassen näher behandelt. Dabei werden für jede Klasse unter Punkt a) die Objekte, unter b) die Relationen, unter c) die Aktivitäten und unter d) die untersuchten Leistungsgrößen beschrieben, die bei der jeweiligen Sichtweise des Systems im Vordergrund stehen und deshalb in den Modellen wiederzufinden sind.

2.4.1 Simulation von Bedienungs-/Wartesystemen

Die Bedienungs-/Wartemodelle stellen die wichtigste Modellklasse in der diskreten Simulation dar. Sie beschreiben Systeme, in denen Kunden oder Aufträge, die bestimmte Bedienungsanforderungen stellen, von Bedienstationen mit einer Anzahl von Bedieneinheiten abgefertigt werden. Beispielsweise kann ein Verkaufsstand als Bedienstation betrachtet werden, wobei die Verkäufer die Bedieneinheiten darstellen. Ein weiteres Beispiel ist ein Rechensystem mit einer bestimmten Anzahl von Druckern. Sollte eine Anforderung zum gegebenen Zeitpunkt nicht erfüllbar sein, werden die Kunden bzw. Aufträge in **Warteschlangen** vor den Bedienstationen eingereiht. Im folgenden wird

häufig nur von Kunden gesprochen; alle Aussagen gelten jedoch sinngemäß auch für Aufträge.

Sowohl Bedienstationen als auch Bedieneinheiten werden gelegentlich als *Ressourcen* bezeichnet. "Ressource" ist jedoch ein allgemeiner Begriff für Hilfsmittel, mit denen Bedürfnisse erfüllt werden können. Dabei kann zwischen *verbrauchbaren* und *nicht verbrauchbaren* Ressourcen unterschieden werden. Die verbrauchbaren Ressourcen werden weiter eingeteilt in *erneuerbare* und *nicht erneuerbare*. Im Zusammenhang mit Bedienungs-/Wartemodellen kommen i. a. nur **nicht verbrauchbare Ressourcen** vor. Es handelt sich um die Bedienstationen und deren Bedieneinheiten, die nach Freigabe durch einen bedienten Kunden wieder zur Verfügung stehen. Im Zusammenhang mit Rechensystemen ist auch die Bezeichnung *Betriebsmittel* gebräuchlich.

Mit Bedienungs-/Wartemodellen werden primär das Warteverhalten von Kunden und die Auslastung von Bedienstationen untersucht. Gesucht ist die günstigste Bedienstrategie unter den (z. T. konkurrierenden) Zielkriterien:

- minimale Anzahl von Bedienstationen
- minimale Wartezeiten
- maximale Auslastung der Bedienstationen

Für bestimmte Standardmodelle von Bedienungssystemen existieren analytische Lösungen. Bei komplexen Bedienungsproblemen empfiehlt sich dagegen eine angenäherte Lösung durch stochastische, zeitdiskrete Simulation der Bedienstationen, sollen keine gravierenden Vereinfachungen in Kauf genommen werden.

2.4.1.1 Systemstruktur

a) Objekte

Die Objekte, aus denen Bedienungs-/Wartemodelle bestehen, sind *nachfragende Instanzen* (Kunden, Aufträge) und *nicht verbrauchbare Ressourcen* (wie Verkaufsstände, Praxisräume, Maschinengruppen usw., allgemein Bedienstationen). Sie sind jeweils durch eine Menge von Attributen charakterisiert. Für die **Kunden** oder **Aufträge** sind dies:

- Ankunftsverteilung
- Servicebedarf
- Präferenzen für eine bestimmte Bedieneinheit
- Ressourcenbedarf
- Wartebereitschaft
- Priorität
- Bedienungsfolge

Die **Bedienstationen** sind durch folgende Merkmale gekennzeichnet:

- Anzahl der Bedieneinheiten
- Leistungsfähigkeit der Bedieneinheiten
- Bedienzeitverhalten
- Warteraum
- Warteschlangendisziplin (Bedienstrategie)
- Serviceunterbrechung

Die aufgeführten Merkmale werden nun ausführlicher dargestellt (vgl. auch [Fis78]).

Merkmale der Kunden/Aufträge

Ankunftsverteilung: Die Ankunftsverteilung eines Kundentyps legt die Zeitpunkte fest, zu denen Objekte dieses Typs in das betrachtete System eintreten. Bei einer deterministischen Ankunftsverteilung wird im einfachsten Fall eine *Zwischenankunftszeit*, also das Zeitintervall zwischen den Ankunftszeitpunkten zweier aufeinanderfolgender Kunden, fest vorgegeben. Es ist aber auch denkbar, den nächsten Ankunftszeitpunkt auf andere Weise als Funktion des vorigen Ankunftszeitpunktes (oder die nächste Zwischenankunftszeit als Funktion der vorigen Zwischenankunftszeit) anzugeben. Im Falle einer stochastischen Ankunftsverteilung sind die Zwischenankunftszeiten i. a. unabhängig identisch verteilt, d. h. sie sind Realisierungen einer Folge untereinander stochastisch unabhängiger Zufallsvariablen, die alle dem gleichen statistischen Verteilungsgesetz folgen. Häufig wird hierfür die **Exponentialverteilung** oder die **Erlang-Verteilung** gewählt (s. 4.1.3).

Weitere Eigenschaften der Ankunftsverteilung sind Anfangszeitpunkt und Endzeitpunkt oder die maximale Zahl von Ankünften. Eine Alternative zur Modellierung der Ankunftsverteilung als Kundenattribut besteht darin, sie als Attribut eines eigenständigen Objekts *Quelle* anzusehen, das gemäß der angegebenen Verteilung Kunden eines bestimmten Typs generiert. Diese Möglichkeit wurde in der Simulationssprache GPSS [Rös78] realisiert. Entsprechend der Quelle für Kunden eines bestimmten Typs wäre dann auch deren Senke anzugeben (siehe auch Punkt "Bedienungsfolge").

Servicebedarf: Die Kunden oder Aufträge können sich durch ihren Servicebedarf (die Art ihrer Anforderungen an die Bedienstationen) unterscheiden. Zum Beispiel haben Jobs in einem Rechensystem unterschiedliche Betriebsmittelforderungen. Der Servicebedarf kann durch Attribute dargestellt werden. Eine andere Möglichkeit besteht darin, für Kunden oder Aufträge mit unterschiedlichen Anforderungen verschiedene Typen von Objekten zu definieren.

Präferenzen für Bedieneinheiten: Es ist möglich, daß die Anforderungen von Kunden nur von bestimmten Bedieneinheiten einer Bedienstation (z. B. Friseur in einem Fri-

seursalon) erfüllt werden können – sei es, daß nur diese eine bestimmte Leistung erbringen können (z. B. eine bestimmte Frisur beherrschen), sei es, daß die Kunden von sich aus bestimmte Präferenzen haben. Im Extremfall könnten alle Kunden nur eine bestimmte Bedieneinheit akzeptieren. Dann könnte die entsprechende Bedienstation mit n Bedieneinheiten auch als System von n Bedienstationen mit jeweils einer Bedieneinheit modelliert werden. Dies ist jedoch nicht möglich, wenn es Kunden ohne Präferenzen gibt, wenn Präferenzen sich auf mehrere Bedieneinheiten beziehen (Frau Koch oder Herr Schmidt) oder wenn negative Präferenzen bestehen (auf keinen Fall eine Auszubildende im ersten Jahr). In diesem Fall müssen im Modell Bedieneinheiten mit individuellen Attributen und positive oder negative Präferenzen für solche Einheiten darstellbar sein.

Ressourcenbedarf: Ein weiterer Fall, der die Modellierung von Anforderungen erschwert, ist gegeben, wenn Kunden für ihre Bedienung bestimmte *Kombinationen* von Bedienstationen gleichzeitig belegen müssen. Zur Verdeutlichung soll das folgende Beispiel dienen:

Bei einer Flotte von Frachtern tritt laufend Bedarf nach Ersatzteilen verschiedenen Gewichts und Volumens auf. Die Ersatzteile müssen per Flugzeug zum jeweiligen Frachter befördert werden, wobei ebenfalls eingeflogene Mechaniker die Reparatur durchführen. Von allen drei Bedienstationen – Flugzeugpark, Pilotenteam und Mechanikerteam – wird also gleichzeitig eine Bedienung gefordert.

Erschwerend könnte hinzukommen, daß je nach Entfernung des Frachters vom Flugplatz nur bestimmte Flugzeuge eingesetzt werden, die auch nur von bestimmten Piloten geflogen werden können. Ebenso könnten für eine gegebene Reparatur nur bestimmte Mechaniker einsetzbar sein. Hier wären also wiederum Präferenzen für bestimmte Bedieneinheiten zu berücksichtigen.

In einem solchen System können sogenannte Verklemmungsprobleme (*Deadlock-Probleme*) auftreten (s. auch 7.10). Im Realsystem wie im Modell muß hierfür eine Lösungsstrategie vorgesehen sein.

Wartebereitschaft: Es kann vorkommen, daß Kunden, die auf eine belegte Bedienstation stoßen, nicht bereit sind, länger als eine bestimmte Frist auf ihre Bedienung zu warten. Sie verlassen also das System ggf. nach einer bestimmten Wartezeit. Derartige Verluste von Kunden oder Aufträgen müssen in der Modellstruktur und der Beurteilung des Modellverhaltens berücksichtigt werden.

Priorität: Bestimmte Kunden oder Aufträge können in einem System mit höherer Priorität behandelt werden als andere (z. B. die Freundin der Chefin im Friseursalon oder ein eiliger Auftrag in einer Werkstatt). Hier ist zu unterscheiden, ob die Prioritäten ausschließlich bei der Verwaltung der Warteschlangen berücksichtigt werden (die Wartenden werden nach Prioritäten sortiert), oder ob höher priorisierte Kunden die

Bedienung anderer Kunden sogar unterbrechen und sie damit aus der Bedieneinheit verdrängen können.

Bedienungsfolge: Häufig benötigt ein Kunde oder Auftrag nicht nur eine Bedienstation, sondern mehrere in Folge. Beispielsweise durchläuft ein Werkstück verschiedene aufeinanderfolgende Bearbeitungsvorgänge; ein Supermarkt-Kunde steht zuerst an der Käsetheke an und später an der Kasse. Jeder Kundentyp hat daher eine spezifische Folge von Bedienstationen, die seinen Weg von der Ankunft im System (Quelle) bis zu seinem Austritt (Senke) beschreibt. Die Bedienungsfolge besagt erstens, welche Bedienstationen des Systems der Kunde durchläuft, und zweitens, wie diese zeitlich geordnet sind.

Von der linearen Bedienungsfolge gibt es drei mögliche Abweichungen. Im ersten Fall sind für einen Kundentyp mehrere alternative Abschnitte von Bedienungsfolgen vorgesehen. Die Entscheidung darüber, welche der möglichen Bedienungsfolgen für einen individuellen Kunden gewählt wird, kann entweder zufällig, aufgrund bestimmter Kundenattribute oder abhängig vom Systemzustand (Länge der Warteschlangen vor den Bedienstationen usw.) erfolgen.

Die zweite Möglichkeit liegt in der Verbindung von Bedienungsfolgen von Kunden verschiedenen Typs. Die Kunden absolvieren einen gemeinsamen Abschnitt ihrer Bedienungsfolge und trennen sich dann wieder. Von jedem der beteiligten Kundentypen kann eine bestimmte Anzahl von Kunden erforderlich sein, damit alle Kunden gemeinsam den Weg fortsetzen können. In diesem Fall besteht zwischen den Kunden eine Synchronisationsbeziehung.

Ein dritter Fall der Abweichung von linearen Bedienungsfolgen ist gegeben, wenn Aufträge in Teilaufträge zerlegt werden, deren Ergebnisse dann an anderer Stelle wieder zusammengeführt werden müssen. Auch hier bestehen Synchronisationsbeziehungen (s. auch Punkt b)).

Merkmale der Bedienstationen

Anzahl und Leistungsfähigkeit der Bedieneinheiten: Dem Servicebedarf, den Präferenzen für bestimmte Bedieneinheiten und dem Ressourcenbedarf auf Kundenseite stehen Anzahl und Leistungsfähigkeit der Bedieneinheiten gegenüber, die eine Bedienstation aufweist.

Bedienzeitverhalten: Wie die Zwischenankunftszeiten können auch die Bedienzeiten deterministisch oder stochastisch verteilt sein. Es ist zu berücksichtigen, daß bei Serviceunterbrechungen evtl. ein Verwaltungsoverhead zur Bedienzeit hinzugerechnet werden muß.

Warteraum: Die Verfügbarkeit eines Warteraums bzw. einer Warteschlange ist eines der wichtigsten Merkmale einer Bedienstation. Es ist zu unterscheiden zwischen Warte-

räumen mit endlicher und solchen mit unendlicher Kapazität. Ist *kein* Warteraum vorhanden, so handelt es sich bei der Bedienstation um ein *Verlustsystem*, d. h. jeder Kunde oder Auftrag, der bei seiner Ankunft nicht sofort bedient werden kann, geht verloren.

Bedienstrategie und Serviceunterbrechung: Die Bedienstrategie entscheidet über die Reihenfolge der Bedienung der Kunden, die sich im Wartraum der Bedieneinheit befinden, unter Berücksichtigung ihrer Priorität. Dabei kann die Bedienung unterbrochen werden (Serviceunterbrechung), z. B. um eintreffende höher priorisierte Kunden sofort zu bedienen. Die aus der Bedienstation verdrängten Kunden werden dann wieder in die Warteschlange eingereiht. Die häufigsten Bedienstrategien für Kunden mit *gleicher* Priorität sind *First In First Out (FIFO)*, *Last In First Out (LIFO)*, *Shortest Job First (SJF)*, *Round Robin (RR)*, d. h. Zeitscheibenstrategie und *Random* (zufällige Auswahl).

b) Relationen

Die Relationen zwischen **Kunden und Ressourcen** wurden als relationale Attribute der Kunden aufgefaßt und deshalb bereits unter Punkt a) behandelt.

Als weitere Relation zwischen Objekten kommt in Bedienungs-/Wartemodellen vor allem die **Synchronisation von Kunden** in Betracht. Eine typische Synchronisationsbeziehung liegt vor, wenn sich an einer bestimmten Stelle eines Bedienungs-/Wartemodells eine bestimmte Anzahl von Kunden ansammeln muß, bevor diese ihren Weg fortsetzen können. Beispielsweise sammeln sich Autos am Anleger einer Fähre, die erst ab einer bestimmten Zahl von Autos übersetzt.

Erheblich komplizierter wird das Problem, wenn Aufträge in teilweise **nebenläufige Teilaufträge** zerlegt werden. Hier besteht diese Synchronisationsbeziehung nicht zwischen beliebigen Aufträgen oder Kunden, sondern zwischen zusammengehörigen Teilaufträgen.

Eine **Synchronisation von Ressourcen** ist ebenfalls denkbar, wenn mehrere Ressourcen zur Bearbeitung eines Auftrages benötigt werden. Im Beispiel des Frachterflottenmodells sind diese Ressourcen die Flugzeuge, Piloten und Mechaniker, die zur Erledigung eines Reparaturauftrages benötigt werden.

c) Aktivitäten

In Bedienungs-/Wartemodellen ist nur eine Aktivität von Bedeutung: die **Bedienung**. Die Bedienzeit wird häufig nur als eine Attribut der Bedienstationen angesehen. Dies ist aber unangemessen, denn die Bedienzeit hängt im allgemeinen sowohl vom Kunden als auch vom Bedienzeitverhalten der Station ab. Die Dauer der Aktivität Bedienung wird somit aus der Kombination von Ressourcen- und Kundenattributen ermittelt.

d) Untersuchte Leistungsgrößen

Als wesentliche Fragestellung von Bedienungs-/Wartemodellen wurden eingangs bereits das Warteverhalten von Kunden und die Auslastung von Bedienstationen genannt. Hauptsächlich interessiert man sich dabei für den Zusammenhang zwischen der Anzahl der Bedienstationen und deren Bedienstrategien einerseits und bestimmten Leistungsgrößen des Systems andererseits. Gemessen werden i. a. Mittelwert und Varianz der folgenden Größen:

- Auslastung der Bedienstation (in Prozent)
- Systemfüllung (Anzahl der Kunden im System)
- Aufenthaltsdauer (Wartezeit plus Bedienzeit)
- Befriedigung der Anforderungen, d. h.
 - Erfüllung von Präferenzen
 - Warteverhalten der Kunden (Wartezeit bzw. Warteschlangenlänge)
 - Verlustwahrscheinlichkeit für jeden Kundentyp

Als *Verlustwahrscheinlichkeit* wird die Wahrscheinlichkeit bezeichnet, mit der ein Kunde das System ohne Bedienung verläßt.

2.4.1.2 Beispiel: Ein Bedienungssystem mit mehreren Servicestellen und begrenztem Warteraum

Systemspezifikation

Es soll die telefonische Platzreservierung einer Bahnlinie als ein spezielles Bedienungssystem mit begrenztem Warteraum untersucht werden. Das zu modellierende System ist folgendermaßen spezifiziert:

1. Es handelt sich ausschließlich um eine telefonische Platzreservierung.
2. Es stehen insgesamt 5 Reservierungskräfte zur Verfügung.
3. Es sind 18 Telefonleitungen geschaltet (d. h. Kapazität des Warteraums gleich 13).
4. Jeder Kunde bucht entweder nur eine einfache oder eine Hin- und Rückfahrt.
5. Wenn alle Reservierungskräfte belegt sind, so ertönt die Ansage "Bitte warten Sie!".
6. Wenn eine Reservierungskraft frei wird, soll derjenige Kunde bedient werden, der schon am längsten wartet.
7. Die Kunden warten nur eine begrenzte Zeit. Die mittlere Wartebereitschaft beträgt 4 Minuten (normalverteilt). Nach Ablauf seiner Wartebereitschaft verzichtet der Kunde endgültig auf die telefonische Reservierung.
8. Die Zwischenankunftszeiten sind exponentialverteilt mit Mittelwert 20 Sekunden, die Servicezeiten sind ebenfalls exponentialverteilt, für eine einfache Fahrt mit Mit-

telwert eine Minute, für eine Hin- und Rückfahrt mit Mittelwert zwei Minuten. Die Wahrscheinlichkeit für eine Hin- und Rückfahrt beträgt 0.75.

Die Systemleistung ist hinsichtlich der folgenden Kriterien zu analysieren:

- mittlere Wartezeit der Anrufer
- Auslastung der Reservierungskräfte
- mittlere Warteschlangenlänge
- Auslastung der Leitungen
- Anteil der sofort bedienten Anrufe
- Anteil der bedienten Anrufe mit Wartezeit
- Anteil der Anrufe, die keine freie Leitung erhalten
- Anteil der Anrufe, die innerhalb der Wartebereitschaft nicht bedient werden

Modellentwurf

Die Zustandsvariablen des Modells sind der Bedienungsstatus einer jeden Reservierungskraft (*frei* oder *beschäftigt*), die Anzahl der wartenden Anrufer in der Warteschlange sowie die Anzahl der freien Leitungen.

Eine erste Strukturierung der Abläufe im spezifizierten System zeigt Abbildung 2-4. Im nächsten Schritt müssen nun **Ereignisse** definiert werden. Wir erhalten sie z. T. durch Zusammenfassung der in Abbildung 2-4 gezeigten Zustandsübergänge:

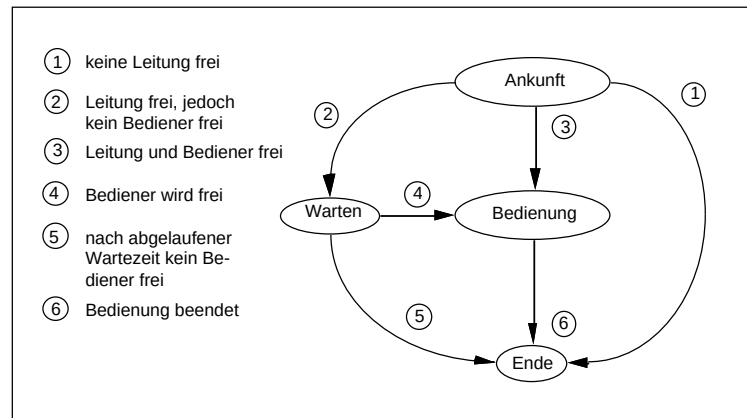


Abb. 2-4: Mögliche Ereignisse bzw. Teilaktivitäten im Reservierungsmodell

1. **Kundenanruf**: Generierung des Anrufs und Zuteilung einer Reservierungskraft oder Einreihen in die Warteschlange, wenn freie Leitung vorhanden.
2. **Bedienungsende**: Freisetzen der Reservierungskraft oder Zuweisung eines neuen Anrufers (falls vorhanden).
3. **Wartenende**: Löschen eines Kunden mit abgelaufener Wartebereitschaft aus der Warteschlange.
4. **Simulationsende**: Ende der Simulation mit Berechnung der Simulationsergebnisse (Reportgenerierung).

Diese informellen Ereignisbeschreibungen sind später zu **Ereignisroutinen** zu präzisieren. Daneben benötigt man, wie in 2.3 erläutert, noch Prozeduren, die Verwaltungsaufgaben übernehmen. Dazu gehören die Zeitföhrungsroutine und der Programmteil zur Aktivierung der jeweils benötigten Ereignisroutine. Diese beiden Programmteile sind hier zur **Prozedur Ereignisauswahl** zusammengefaßt. Standardmäßig werden auch die **Initialisierungsprozedur** und die **Reportprozedur** benötigt.

Der Vorgang der Bedienung kommt in zwei Ereignisroutinen vor: In **Kundenanruf** wird der jeweilige Kunde sofort bedient, falls eine Reservierungskraft frei ist. In **Bedienungsende** wird ein weiterer Kunde bedient (falls vorhanden), nachdem der aktuelle das System verlassen hat. Aus diesem Grunde wird eine **Prozedur Bedienung** vorgesehen.

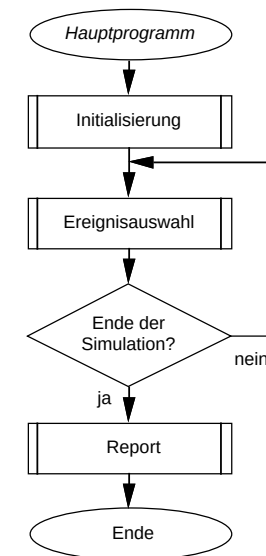
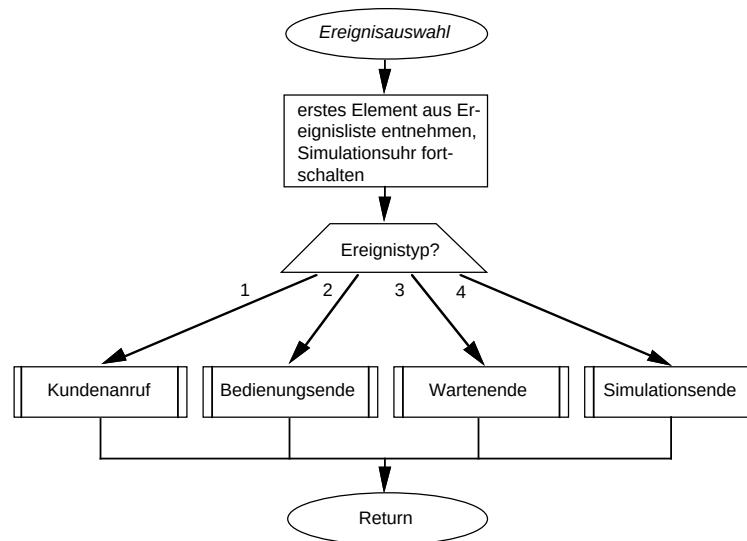


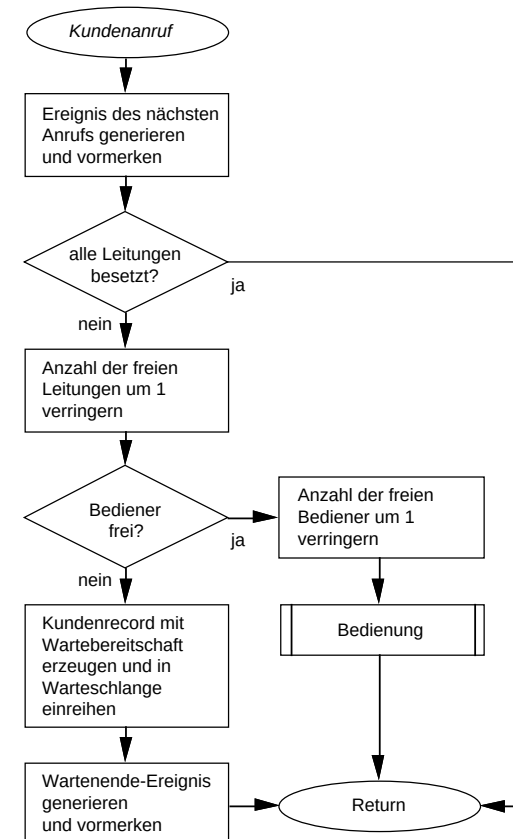
Abb. 2-5: Flußdiagramm des Hauptprogramms

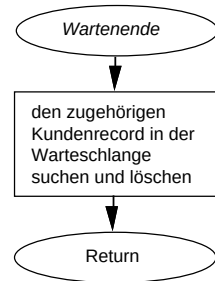
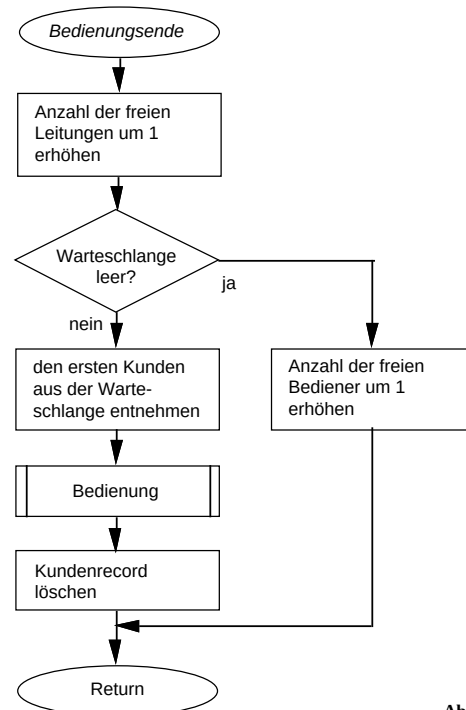
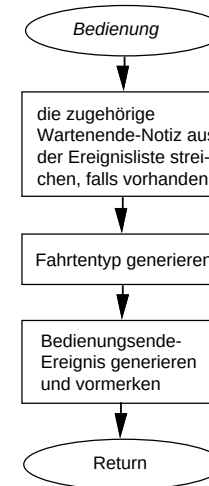
Das Hauptprogramm (Abb. 2-5) hat aufgrund der Einführung der Prozedur *Ereignisauswahl* eine einfache Struktur. Diese Prozedur (Abb. 2-6) entspricht im wesentlichen der in 2.3 definierten Zeitführungsroutine, erledigt hier aber zusätzlich auch den Aufruf der jeweiligen Ereignisroutine. Ereignistyp 4 (Simulationende) führt nach Rückkehr in das Hauptprogramm zur Ergebnisauswertung und -ausgabe.

Beim Aufruf der Ereignisroutine *Kundenanruf* (Abb. 2-7) wird unmittelbar das nächste Ereignis des gleichen Typs generiert und in die Ereignisliste eingetragen. Hier wird also die bereits erwähnte Technik verwendet, eine Folge gleichartiger Ereignisse dadurch zu generieren, daß jedes Ereignis des entsprechenden Typs seinen Nachfolger erzeugt und vormerkt. Findet ein Kunde keine freie Leitung vor (ist die Kapazität des Warteraumes also erschöpft), so endet die Ereignisliste ohne weitere Aktionen. Ist eine Bedienungskraft unbeschäftigt, erfolgt der Aufruf der Prozedur *Bedienung*. In der Ereignisroutine *Kundenanruf* wird aus Vereinfachungsgründen nur dann ein Kundenrecord generiert, wenn dieser nicht sofort bedient, sondern in eine Warteschlange eingetragen wird. Denn nur für diesen Fall muß einem Kunden überhaupt ein Attribut mit der nach der vorgegebenen Normalverteilung generierten Wartebereitschaft zugewiesen werden. Für den Fall der sofortigen Bedienung werden allein die entsprechenden statistischen Zähler aktualisiert (Dekrementierung der Anzahl der freien Bediener).

Abb. 2-6: Prozedur *Ereignisauswahl*

Die Ereignisroutine *Wartenende* (Abb. 2-8) bedarf keiner Erläuterung. Das Ende der Bedienung (Abb. 2-9) führt dazu, daß eine Leitung freigegeben wird und, falls noch Kunden auf anderen Leistungen warten, deren erster bedient wird. Anschließend wird der Kundenrecord gelöscht. (Dies kann bei Simulation einer größeren Anzahl von Kunden aus Speicherplatzgründen notwendig sein.) In der Prozedur *Bedienung* (Abb. 2-10) wird stochastisch der *Fahrtentyp* (einfache Fahrt oder Hin- und Rückfahrt) mit jeweils unterschiedlichen Bedienzeiten generiert.

Abb. 2-7: Ereignisroutine *Kundenanruf*

Abb. 2-8: Ereignisroutine *Wartenende* (Ende der Wartebereitschaft)Abb. 2-9: Ereignisroutine *Bedienungsende*Abb. 2-10: Prozedur *Bedienung*

2.4.1.3 Implementation in Modula-2

Das oben skizzierte ereignisorientierte Simulationsmodell wurde in der Programmiersprache Modula-2 implementiert. Der Quelltext ist im Anhang B.1 wiedergegeben und wird im folgenden erläutert.

Programmaufbau

Das Programm besteht aus vier Modulen, dem Hauptmodul *Reservation* und den Modulen *FixedPointIO*, *Distribution* und *LongReal*. *FixedPointIO* exportiert die Prozedur *WriteFixedPoint*, die eine *REAL*-Zahl in Festkomma-Darstellung ausgibt. In *Distribution* befinden sich die Funktionen für die **Zufallszahlenerzeugung** und die Verteilungen *Exponential* und *Norm*. Die Funktion *LongToReal* in *LongReal* wandelt eine *LONGINT*-Zahl in eine *REAL*-Zahl um. Der Typ *CARDINAL* ist für die Darstellung der **Simulationszeit** zu klein, deshalb wurde *LONGINT* verwendet. Die Simulationszeit wird in Sekunden gemessen. Die folgenden Ereignisse mit entsprechenden **Ereignisroutinen** sind definiert:

- Kundenanruf (*Call*)
- Bedienungsende (*Completion*)
- Wartenende (*MaxWaitet*)
- Simulationsende (*EndofSimulation*)

Es werden zwei dynamische Datenstrukturen definiert: die **Ereignisliste** und die **Warteschlange** für wartende Kunden. In der Ereignisliste *EvTOP* sind die zukünftigen Ereignisse nach Ereigniszeitpunkt und Priorität geordnet. Das Ereignis *Bedienung* besitzt die höchste, *Anruf* die niedrigste Priorität. Das jeweils nächste Ereignis wird von der Prozedur *SelectEvent* ausgewählt, die vom Hauptprogramm solange wiederholt aufgerufen wird, bis die Simulationszeit abgelaufen ist.

Die im Programm verwendeten globalen Variablen sind:

<i>EvTOP</i>	Ereignisliste
<i>QuTOP</i>	Warteschlange für die Anrufer
<i>CurrentTime</i>	aktuelle Simulationszeit
<i>SumOfServTimes</i>	Summe der Bedienzeiten
<i>SimulationTime</i>	Dauer der Simulation
<i>MeanServiceTime</i>	mittlere Bedienzeit für eine einfache Fahrt
<i>MeanIntercallTime</i>	mittlere Zwischenankunftszeit
<i>Seed</i>	Anfangswert für den Zufallszahlenstrom
<i>MeanWaitingTime</i>	mittlere Wartebereitschaft eines Kunden
<i>Deviation</i>	Abweichung von der mittleren Wartebereitschaft
<i>Server</i>	Anzahl der freien Bediener
<i>Lines</i>	Anzahl der freien Telefonleitungen
<i>ImmedServed</i>	Anzahl der sofort bedienten Kunden
<i>NoLine</i>	Anzahl der Anrufe ohne freie Leitung
<i>NoServer</i>	Anzahl der Anrufe ohne Bedienung
<i>Served</i>	Anzahl der bedienten Anrufer

Im folgenden werden die einzelnen Prozeduren des Moduls *Reservation* kurz erläutert.

Initialisierung: In der Prozedur *Initialization* werden die Variablen initialisiert bzw. interaktiv vom Terminal eingelesen. Danach wird das erste Kundenanruf-Ereignis in die Ereignisliste *EvTOP* eingetragen.

Ereignisroutine Anruf: Die Prozedur *Call* generiert zunächst einen neuen Anruf und ordnet ihn in die Ereignisliste ein. Wenn keine Telefonleitung frei ist, wird *Call* verlassen. Sonst wird die Anzahl der freien Leitungen um 1 reduziert. Wenn ein Bediener frei ist, wird auch die Anzahl der Bediener um 1 reduziert und die Prozedur *Service* aufgerufen. Da die Bedienung hier unmittelbar zum Ankunftszeitpunkt des Kunden begonnen wird, muß kein Ereignis *Wartenende* vorgemerkt werden. Außerdem ist auch kein Kundenrecord erforderlich, denn ein Eintritt in die Warteschlange ist für diesen Kunden ausgeschlossen. Andernfalls muß für den Anrufer ein Kundenrecord erzeugt und in die Warteschlange eingereiht werden. Schließlich werden eine Wartebereitschaft generiert und das Ereignis *Wartenende* in der Ereignisliste für den errechneten Zeitpunkt vorge-merkt, und zwar hinter dem Ereignis *Bedienungsende*, wenn es zeitgleich ist, jedoch vor allen anderen zeitgleichen Ereignissen.

Ereignisroutine Wartenende: Die Prozedur *MaxWaited* sucht in der Warteschlange *QuTOP* nach dem Kunden, dessen Summe aus Ankunftszeit und maximaler Wartezeit gleich der aktuellen Simulationszeit ist. Dieser Kunde wird dann gelöscht und seine Leitung wird freigegeben.

Ereignisroutine Bedienungsende: Die Prozedur *Completion* gibt zunächst eine Leitung frei. Danach wird geprüft, ob die Warteschlange leer ist. Ist dies der Fall, wird die Zahl der freien Bediener um 1 erhöht. Andernfalls wird der nächste Anrufer aus der Warteschlange geholt, die Prozedur *Service* aufgerufen und der zugehörige Kundenrecord gelöscht.

Ereignisroutine Simulationsende: Die Prozedur *EndofSimulation* ruft einzig die Auswertungsprozedur *Report* zum Ende der Simulationszeit auf.

Weitere Prozeduren: Die Prozedur *DeleteMaxWaited* dient dazu, das Wartenende-Ereignis des Kunden, dessen Bedienung gerade beginnt, aus der Ereignisliste zu entfernen. Ein Ende der Wartebereitschaft kann für diesen Kunden ja nicht mehr eintreten. Die Bedienungsprozedur *Service* generiert die Bedienzeit mit Hilfe einer Exponentialverteilung, deren Parameter (der Mittelwert) abhängig vom Fahrkartentyp (Hin- oder Hin- und Rückfahrt) bestimmt wird. Die Funktion *Type* liefert diesen Mittelwert, wobei auf den Zufallsgenerator *Random* zurückgegriffen wird. Anschließend generiert *Service* eine Ereignisnotiz für das zugehörige Bedienungsende-Ereignis und trägt diese in die Ereignisliste ein. Das Einordnen der Ereignisse in die Ereignisliste *EvTOP* gemäß Zeitpunkt und Priorität geschieht durch die Prozedur *ScheduleEvent*. Die Prozedur *SelectEvent* entnimmt das vorderste Ereignis aus der Ereignisliste, schaltet die Simulationsuhr auf den nächsten Ereigniszeitpunkt und ruft die zugehörige Ereignisroutine auf. *Report* schließlich errechnet die Statistik und schreibt die Ein- und Ausgabedaten in einem entsprechenden Tabellenformat in eine Datei.

Hauptprogramm: Im Hauptprogramm wird anfangs die Initialisierungsroutine aufgerufen und dann die Prozedur *SelectEvent* bis zum Simulationsende wiederholt ausgeführt. Der Aufruf der Reportprozedur erfolgt hier nicht im Hauptprogramm, sondern in der Ereignisroutine *EndofSimulation*.

Die Prozeduraufrufe im Hauptmodul *Reservation* des Modula-2-Simulationsprogramms sind wie folgt verschachtelt:

```

MODULE Reservation
BEGIN
  1 Initialization
    2 ScheduleEvent
  1 SelectEvent
    2 Call
      3 ScheduleEvent
      3 Service
        4 ScheduleEvent
      3 ScheduleEvent
    2 Completion
      3 Service
        4 DeleteMaxWaited
        4 ScheduleEvent
    2 MaxWaited
    2 EndofSimulation
    3 Report
END Reservation

```

Simulationsergebnisse

Mit dem Modell wurde eine Simulation für die Dauer von 100 Stunden (= 360000 Sekunden) durchgeführt. In dieser Simulationszeit wurden ca. 18000 Kundenanrufe generiert, von denen ca. 15800 auch bedient wurden. Nur 161 Anrufe hatten keine freie Leitung vorgefunden, und ca. 2160 Anrufer gaben nach abgelaufener Wartebereitschaft vorzeitig auf. Die durchschnittliche Wartezeit betrug ca. 70 Sekunden, wobei auch Anrufer mit sofortiger Bedienung berücksichtigt sind. Ohne die sofort bedienten Kunden ergab sich für die wartenden Anrufer eine durchschnittliche Wartezeit von über 90 Sekunden. Die **Auslastungsstatistiken** wurden wie folgt berechnet:

$$\text{Mittlere Auslastung eines Reservierers} = \frac{\sum \text{Bedienzeiten}}{5 \cdot \text{Simulationszeit}}$$

$$\text{Mittlere Auslastung der Leitungen} = \frac{\sum (\text{Bedienzeiten} + \text{Wartezeiten})}{18 \cdot \text{Simulationszeit}}$$

Es ergab sich eine Auslastung von ca. 91% für die Bediener und ca. 45% für die Telefonleitungen. Die Simulationsergebnisse sind in Form des Simulations-Reports in Anhang B.1 angegeben. Diese Ergebnisse können jedoch nicht als statistisch gesichert angesehen werden, da **keine statistische Genauigkeitsabschätzung** vorgenommen wurde (s. hierzu 4.4.3).

2.4.2 Simulation von Lagerhaltungssystemen

Einen weiteren wichtigen Modelltyp der zeitdiskreten Simulation bilden die Lagerhaltungsmodelle. *Lager* dienen als Puffer zwischen Produktion (oder Anlieferung) und Verbrauch. Modelle der Lagerhaltung beschreiben den Betrieb eines Lagers mit einer Anzahl von Produkten, die entsprechend dem Eingang von Aufträgen aus dem Lager entfernt und beim Eintreffen von bestellten Lieferungen wieder aufgefüllt werden. Wenn ein Auftrag nicht sofort (vollständig) erfüllt werden kann, werden (Rest-) Aufträge zurückgestellt. Obwohl es viele verschiedene Arten von Lagern gibt, z. B. Produktions-, Verkaufs- oder Zentrallager, haben sie dennoch die Hauptziele der Lagerhaltung gemein. Es bestehen nur Unterschiede in der *Bewertung* der einzelnen Ziele. Die Hauptziele der Lagerhaltung sind:

1. Hohe Deckung der Nachfrage (Sofortlieferefähigkeit)
2. Niedrige Lagerhaltungskosten (laufende Kosten für die Lagerhaltung)
3. Niedriger Kapitaleinsatz (Lagerwert und Investitionen für das Lager)

Diese Ziele stehen z. T. in Konkurrenz zueinander. Eine hohe Deckung der Nachfrage verlangt beispielsweise einen hohen durchschnittlichen Lagerbestand, während ein niedriger Kapitaleinsatz nur durch einen niedrigen durchschnittlichen Lagerbestand erreicht wird. Die Ziele müssen deshalb bewertet und aufeinander abgestimmt werden. So kann man z. B. einen bestimmten Deckungsgrad der Nachfrage vorgeben und dazu die niedrigsten Gesamtkosten für das Lager bei einem beliebigen Kapitaleinsatz suchen. Eine gute Lagerhaltung wird nur dann erreicht, wenn die Ziele gut aufeinander abgestimmt sind. Die wichtigsten Leistungsgrößen bei Lagerhaltungsmodellen sind die Kosten der Lagerhaltung und die Erfüllbarkeit der Aufträge bei verschiedenen Lagerkapazitäten und Bestellpolitiken.

Zwischen den oben besprochenen Bedienungs-/Wartesystemen und Lagerhaltungssystemen gibt es zwei wesentliche Unterschiede. Erstens sind die **Ressourcen** eines Lagerhaltungssystems im Gegensatz zu denen eines Bedienungs-/Wartesystems sowohl verbrauchbar als auch erneuerbar. Zweitens liegt das Interesse bei der Analyse von Lagerhaltungssystemen mehr auf der **sofortigen Erfüllbarkeit** von Aufträgen (Sofortlieferefähigkeit) als auf der Aktivität der Auftragsabfertigung (Bedienung). Die Bedienzeit wird daher in Lagerhaltungsmodellen i. a. nicht modelliert. Dagegen spielen die Warteerfordernisse von Kunden bei nicht sofort erfüllbaren Aufträgen eine ähnlich wichtige Rolle wie bei Bedienungs-/Wartesystemen, wobei aber nicht der Wartevorgang an sich von Interesse ist. Das Ziel einer hohen Sofortlieferefähigkeit bedingt auch eine Minimierung der Kundenwartezeiten.

Lagerhaltungsmodelle lassen sich ohne weiteres mit Bedienungs-/Wartemodellen **kombinieren**. Zum Beispiel ließe sich in dem oben erwähnten Modell einer Frachterflotte eine Lagerhaltungskomponente für Ersatzteile einfügen. Schließlich gibt es Sy-

steme, die sich sowohl durch Bedienungs-/Wartemodelle als auch durch Lagerhaltungsmodelle beschreiben lassen (vgl. hierzu [Sch84]).

Unter bestimmten restriktiven Annahmen – z. B. hinsichtlich der Nachfrageverteilung – existieren für einige Lagerhaltungsmodelle **analytische Lösungen**. Die Analyse komplexerer Lagerhaltungsmodelle ist dagegen nur durch Simulation möglich.

2.4.2.1 Systemstruktur

a) Objekte

Das wichtigste Objekt in einem Lagerhaltungsmodell ist das eigentliche **Lager**, d. h. eine Menge von verschiedenen Produkten oder Artikeln, wobei für jeden Produkttyp ein aktueller Bestand angegeben werden kann. Wie eine Bedienstation bei Bedienungs-/Wartemodellen ist das Lager ein Objekt, das Ressourcen zur Erfüllung bestimmter Aufträge und einen Warteraum für die Aufträge bereitstellt. Es handelt sich konzeptuell um einen Objekttyp, wobei in einem Lagerhaltungsmodell aber häufig nur eine Instanz dieses Typs, also nur ein einziges Lager, vorkommt.

Die weiteren Objekte eines Lagerhaltungsmodells sind Kundenaufträge, die Ressourcen anfordern, Sendungen an Kunden, die den Lagerbestand reduzieren, Bestellungen an einen Lieferanten und Lieferungen, die den Bestand auffüllen.

Merkmale des Lagers

Das Lager enthält, wie bereits erwähnt, die Menge der Produkte sowie einen Warteraum für nicht sofort erfüllbare Aufträge, die sogenannte *Vorbestell-Liste*. Außerdem fallen im Lager verschiedene Arten von Kosten an. Die verschiedenen Produkttypen, die in einem Lager vorgehalten werden, werden im folgenden als *Artikel* bezeichnet.

Bestand: Der aktuelle Bestand eines Artikels ändert sich durch Sendungen und Lieferungen. Bei der Definition eines Lagerhaltungsmodells muß für bestimmte Bestellpolitiken (s. u.) angegeben werden, wie groß der Mindestbestand jedes Artikels sein soll. Bei der Aufgabe von Bestellungen ist außerdem ein maximal möglicher Bestand zu berücksichtigen.

Warteraum und Bedienstrategie: Hier gilt im wesentlichen das gleiche, was über die Warteräume und Bedienstrategien bei Bedienungs-/Wartemodellen gesagt wurde (2.4.1.1). Die Kapazität des Warteraums eines Lagers kann begrenzt oder unbegrenzt sein, oder es kann überhaupt kein Warteraum vorgesehen sein (Kapazität 0).

Kosten: Die Kosten eines Lagers sind für die Modellstruktur selbst nicht von Bedeutung. Sie spielen erst im Zusammenhang mit den angelegten **Zielkriterien** eine Rolle. Man unterscheidet u. a. die folgenden Kostenarten:

- Lagerhaltungskosten
 - Lagerstättenmiete
 - Betriebskosten
 - Personalkosten
 - Versicherungen
 - Kapitalbindung
- Fehlmengenkosten
 - Kosten für zusätzliche Verwaltungstätigkeit
 - Verlust an Kundenvertrauen
 - gegebenenfalls Kosten für außerplanmäßige Beschaffung

Merkmale der Kundenaufträge

Die Aufträge in einem Lagerhaltungsmodell haben zum Teil ähnliche Merkmale wie die Kunden oder Aufträge in einem Bedienungs-/Wartemodell: **Ankunftsverteilung, Artikelbedarf, Wartebereitschaft** und **Priorität** (vgl. 2.4.1.1).

Bei den Präferenzen für bestimmte Artikel und bei der Bedienungsfolge liegen die Dinge jedoch anders. Mit einer *Präferenz für bestimmte Artikel* ist bei Lagerhaltungsmodellen nicht gemeint, daß nur bestimmte Artikel angefordert werden – dies ist bei Warenbestellungen völlig natürlich. Gemeint ist vielmehr, daß für einen Artikel ein oder mehrere andere Artikel genannt werden, die geliefert werden sollen, wenn der erstgenannte Artikel nicht vorrätig ist. Das bedeutet, daß innerhalb einer Teilmenge der angebotenen Artikel bestimmte Artikel präferiert werden. Dies kann etwa dann sinnvoll sein, wenn ein Einzelhändler für einen vergriffenen Artikel wenigstens einen Ersatzartikel anbieten will. Die *Bedienungsfolge* tritt bei Lagerhaltungsmodellen nur noch in degenerierter Form auf, da nur eine einzige Bedienstation (das Lager) vorhanden ist.

Merkmale der Warensendungen

Der Begriff *Warensendung* bezeichnet hier die Auslieferung von Waren, die ein Kunde beim Lager in Auftrag gegeben hat. Der Vorgang der Warensendung wird in Lagerhaltungsmodellen häufig nicht abgebildet (bis auf die Tatsache, daß der Bestand entsprechend reduziert wird). Eine detailliertere Modellierung kann aber durchaus sinnvoll sein. Einerseits entstehen auch für Sendungen Kosten (Verpackung, Transport), die in die Gesamtkosten der Lagerhaltung mit eingehen. Andererseits entspricht in realen Lagern eine Sendung aufgrund von Irrtümern oder Fehlern nicht immer dem ursprüngli-

chen Auftrag. Dies kann dann zu zusätzlichen Aufträgen und Reklamationen führen, die weitere Kosten verursachen.

Eine Warensendung ist charakterisiert durch ihren **Zeitpunkt** und die **Anzahl der ausgehenden Einheiten pro Artikel**. Die Zeitpunkte, zu denen die Sendungen das Lager verlassen, sind nicht unabhängig identisch verteilt, sondern abhängig von der Ankunft der sie auslösenden Aufträge, variiert durch eine stochastische Komponente.

Merkmale der Bestellungen

Der Begriff *Bestellung* bezieht sich hier stets auf eine Aktivität des Lagers mit dem Ziel, den Bestand aufzufüllen.

Zeitliche Verteilung: Wie die Kundenaufträge folgen auch die Zeitabstände zwischen den Bestellungen durch das Lager einer bestimmten Verteilung. Man spricht hier auch vom *Bestellzyklus* des Lagers. Die Intervalle zwischen den Bestellzeitpunkten sind jedoch im allgemeinen nicht unabhängig identisch verteilt. Häufig wird eine deterministische Gleichverteilung verwendet, d. h. man geht davon aus, daß Bestellungen in konstanten Zeitabständen erfolgen (fixe Bestellpunkte). Üblich sind daneben auch variable Zeitabstände, d. h. es wird immer dann bestellt, wenn der Lagerbestand eines Artikels eine bestimmte Grenze unterschreitet.

Bestellmenge: Ebenso wie der Bestellzyklus kann die Bestellmenge eines Artikel konstant oder variabel sein. Dies hängt von der jeweiligen Bestellpolitik ab (s. u.).

Kosten: Für eine Bestellung ist zunächst ein bestimmter Fixkostenbetrag anzusetzen. Darüber hinaus entstehen variable Kosten in Höhe der Bestellmenge, multipliziert mit dem jeweiligen Stückpreis.

Merkmale der Lieferungen

Zeitliche Verteilung: Wie die Warensendungen und die Bestellungen sind auch die Lieferungen an das Lager nicht unabhängig identisch verteilt. Der Zeitpunkt ihres Eintreffens ist abhängig von der Aufgabe ihrer Bestellung und wird i. d. R. durch eine stochastische Komponente variiert.

Kosten: Die Kosten, die als Merkmal der Bestellung erwähnt wurden, können bei einer detaillierteren Modellierung auch der Lieferung zugeordnet werden. Tatsächlich fallen sie ja zum größten Teil frühestens bei Eintreffen der Lieferung an. Der Gedanke liegt nahe, zwischen Bestellmenge und Liefermenge zu unterscheiden, um etwa Fälle berücksichtigen zu können, in denen eine Lieferung falsch, unvollständig oder beschädigt eintrifft. Bei üblichen Lagerhaltungsmodellen wird jedoch von der Annahme ausgegangen, daß Bestell- und Liefermenge identisch sind.

b) Relationen

In einem Lagerhaltungsmodell lassen sich zwei Arten von Beziehungen zwischen den oben beschriebenen Objekten ausmachen. Zum einen bestehen Kausalbeziehungen zwischen verschiedenen Ereignissen: Warensendungen erfolgen aufgrund von Aufträgen, Lieferungen aufgrund von Bestellungen. Für die Modellbildung relevant ist hierbei die zeitliche Verzögerung (*delivery lag*), mit der die Sendung oder Lieferung eintritt. Zum anderen bestehen Beziehungen zwischen Ereignissen und dem Zustand des Lagers: Sendungen vermindern den Bestand, Lieferungen vergrößern ihn, Bestellungen werden durch Differenzen zwischen Ist- und Sollbestand ausgelöst.

c) Aktivitäten

Die **Aufgabe von Bestellungen** ist die wichtigste Aktivität in einem Lagerhaltungssystem. Die Strategie, mit der Zeitpunkt und Umgang der Bestellungen festgelegt werden, wird als *Bestellpolitik* bezeichnet. Die Bestellpolitik ist das Kernstück jedes Lagerhaltungsmodells, denn sie ist es, deren Kosteneffizienz i. d. R. durch die Modellstudie überprüft werden soll.

Die Bestellpolitik legt fest, in welchen Zeitabständen t welche Produktmenge q bestellt werden soll. Da jede der Größen q und t konstant oder variabel gehalten werden kann, ergeben sich vier grundlegende Typen von Bestellpolitiken:

- (q,t)-Politik: Bestellung einer konstanten Bestellmenge q für jeden Artikel in konstanten Zeitabständen t (fixe Bestellpunkte).
- (S,t)-Politik: Die Bestände werden in konstanten Zeitabständen bis zu gegebenen Sollbeständen S aufgefüllt, d. h. es ist $q = S - I$ mit I = Istbestand (für einen gegebenen Artikel).
- (q,s)-Politik: Wenn der Bestand eines Artikels eine Untergrenze s unterschreitet, wird eine konstante Artikelmenge q bestellt.
- (s,S)-Politik: Unterschreitet der Bestand eines Artikels eine Untergrenze s , wird der Bestand bis zu einer Sollmenge S aufgefüllt.

Bei den letzten beiden Bestellpolitiken muß zusätzlich angegeben werden, in welchen Zeitabständen t der Lagerbestand *überprüft* werden soll. Außerdem können sie dahingehend variiert werden, daß aus Anlaß der Bestellung eines Artikels einige weitere Artikel mitbestellt werden, deren Bestand bereits relativ nahe an der Untergrenze liegt.

d) Untersuchte Leistungsgrößen

Die wichtigsten Leistungsgrößen eines Lagers sind die **Sofortlieferfähigkeit** und die aufzuwendenden **Kosten**. Untersucht wird primär die Abhängigkeit dieser Größen von

den Faktoren **Lagerkapazität** und **Bestellpolitik**. Die hauptsächlichsten Kostenarten, die dabei eine Rolle spielen, sind

- Lagerhaltungskosten (vgl. Punkt a), Unterabschnitt "Kosten")
- Fehlmengenkosten (vgl. Punkt a), Unterabschnitt "Kosten")
- Versandkosten
 - Verpackung
 - Transport
- Bestellkosten
 - Fixkosten
 - Stückkosten

Welche von diesen Kostenfaktoren in einem gegebenen Modell berücksichtigt werden müssen und wie sie berechnet werden, ist sehr unterschiedlich. Beispielsweise kann der Transport zu Lasten der Kunden gehen. Auf der anderen Seite können zusätzliche Kosten durch Reklamationen entstehen. Die konkrete Berechnung der Gesamtkosten ist daher modellspezifisch.

2.4.2.2 Beispiel: Ein Mehrprodukt-Lagerhaltungssystem mit Vorbestellungen

Systemspezifikation

Es soll ein Lagerhaltungssystem mit drei verschiedenen Artikeln i ($i \in \{1,2,3\}$) analysiert werden mit dem Ziel, den Sollbestand für jeden Artikel in den nächsten n Monaten festzusetzen. Die Zwischenankunftszeiten der Kundennachfragen seien exponentialverteilt mit einem Erwartungswert von 0.1 Monaten. Die Nachfragemengen D_i seien unabhängige Zufallsvariablen mit den folgenden diskreten Wahrscheinlichkeitsfunktionen:

Nachfragemenge D_i	Wahrscheinlichkeiten $P_i(D_i)$		
	Produkt: $i=1$	$i=2$	$i=3$
0	0	1/6	1/6
1	1/6	0	1/6
2	1/3	1/6	0
3	1/3	1/3	1/6
4	1/6	0	1/3
5	0	1/6	1/6
6	0	1/6	0

Bei jeder Kundenankunft tritt eine Nachfrage bezüglich mehrerer Produkte auf, wobei die Nachfragemengen für die einzelnen Produkte voneinander unabhängig sind. Sollte

eine Kundennachfrage nicht unmittelbar aus dem Lagerbestand befriedigt werden können, so wird sie als Vorbestellung aufgenommen und nach Lieferung neuer Ware erfüllt.

Bezüglich jeder einzelnen Produktart i wird wie folgt vorgegangen: Zu Beginn eines jeden Monats wird der aktuelle Lagerbestand überprüft und entschieden, ob und wieviele Einheiten vom Lieferanten bestellt werden sollen. Wenn Z Einheiten bestellt werden, so entstehen Kosten in Höhe von $K + J \cdot Z$, wobei K die fixen und J die variablen Bestellkosten darstellen. Es gelte $K = 32$ GE (Geldeinheiten) und $J = 3$ GE für alle Artikel. Bei Aufgabe einer Bestellung ist mit einer stochastischen Lieferzeit zu rechnen, die einer Gleichverteilung im Intervall 0,5 bis 1 Monat folgt.

Das Unternehmen bedient sich einer (s, S) -Politik, um die Bestellmengen Z zu bestimmen, also

$$Z = \begin{cases} S-L, & \text{wenn } L \leq s \\ 0, & \text{wenn } L > s \end{cases}$$

wobei L den Lagerbestand zu Beginn eines Monats (vor Bestellung) bezeichnet. Die Werte für s und S werden unten näher spezifiziert. Wenn eine Nachfrage auftritt, wird diese sofort befriedigt, sofern der Lagerbestand größer oder gleich der Nachfragemenge ist. Liegt die Nachfragemenge jedoch *über* dem aktuellen Bestand, so wird die offene Nachfragemenge für eine spätere Auslieferung vermerkt. In diesem Fall wird der **buchmäßige Lagerbestand als negativ verzeichnet**. Kommt eine neue Lieferung an, so werden zuerst die ausstehenden Bestellungen befriedigt; nur der Überschuß wird in das Lager übernommen.

Sei $L(t)$ der buchmäßige Lagerbestand zum Zeitpunkt t (positiv, 0 oder negativ), wobei

$$L^+(t) = \max \{L(t), 0\}$$

den tatsächlichen Lagerbestand zum Zeitpunkt t und

$$L^-(t) = \max \{-L(t), 0\}$$

die Vorbestellmenge aufgrund nicht befriedigter Nachfrage bezeichnen ($L^-(t) \geq 0$). Eine mögliche Realisierung von $L(t)$, $L^+(t)$ und $L^-(t)$ für einen Artikel ist in Anlehnung an [Law82] in Abbildung 2-11 dargestellt. Die durchschnittlichen Lagerhaltungskosten für n Zeitperioden lassen sich bestimmen aus:

$$LK = h / n \int_0^n L^+(t) dt$$

wobei $h = 1$ GE die Lagerhaltungskosten je Artikel im Monat bezeichnet. Die Fehlmengenkosten berechnen sich entsprechend zu

$$FK = j / n \int_0^n L^-(t) dt$$

wobei $j = 5$ GE die Fehlmengenkosten pro Artikel im Monat bedeutet. Die obigen Ausführungen gelten für die anderen beiden Produktarten dieses Beispiels analog.

In der Beispielsimulation werden ein Anfangslagerbestand von $L^+(0) = 60$ und Vorbestellungen $L^-(0) = 0$ für jede Produktart angenommen. Die Simulationszeit wird auf $n = 120$ Monate festgelegt, und die durchschnittlichen Gesamtkosten (Summe aus Bestell-, Lagerhaltungs- und Fehlmengenkosten) dienen als Maß für die Systemleistung zum Vergleich der folgenden vier Bestellpolitiken:

	1	2	3	4
s :	20	20	40	60
S :	60	100	100	80

Aus Einfachheitsgründen seien die Werte für s und S jeweils für alle Produktarten gleich. Anhand der Simulationsergebnisse soll die günstigste dieser vier speziellen (s, S) -Bestellpolitiken ausgewählt werden.

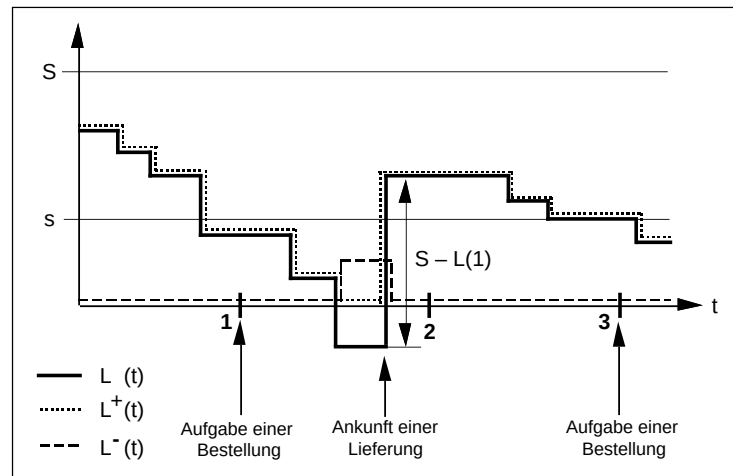


Abb. 2-11: Eine Realisierung von $L(t)$, $L^+(t)$ und $L^-(t)$ im Zeitverlauf

Modellentwurf

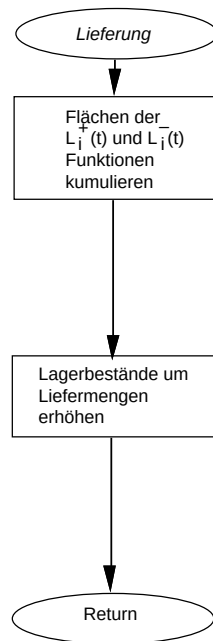
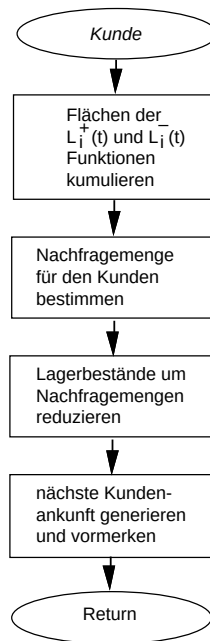
Das Modell des skizzierten Lagerhaltungssystems kann nach dem Modellierungskonzept der ereignisorientierten zeitdiskreten Simulation am einfachsten realisiert werden. Es werden in dem Modell die folgenden **Ereignistypen** definiert:

- *Lieferung*: Ankunft einer Lieferung im Lager
- *Kundenauftrag*: Eingang des Auftrags eines Kunden (bzgl. der drei Produktarten)
- *Simulationsende*: Ende der Simulation nach n Monaten
- *Bestellung*: Lagerbestandsprüfung für jede Produktart am Beginn eines Monats und Bestellung, falls der Lagerbestand kleiner oder gleich s ist.

Die **Zeitführungsroutine** des zu erstellenden Simulationsprogramms stellt sicher, daß am Simulationsende nach n Monaten die Simulation abgebrochen wird, bevor eine neuerliche Bestellung mit zusätzlichen Kosten aufgegeben wird (*Simulationsende* hat höhere Priorität als *Bestellung*), um eine Verfälschung der Simulationsergebnisse zu vermeiden.

In diesem Beispiel wird auf die Angabe eines Quellprogramms verzichtet. Im folgenden wird jedoch aufgezeigt, welche Prozeduren ein entsprechendes Programm enthalten müßte:

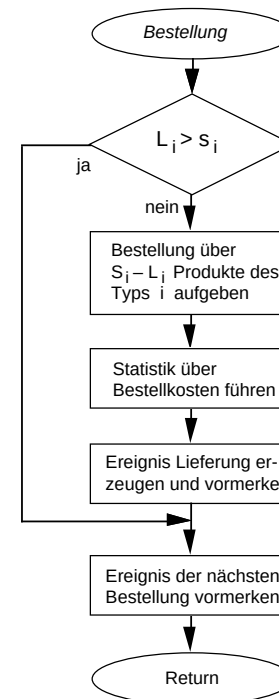
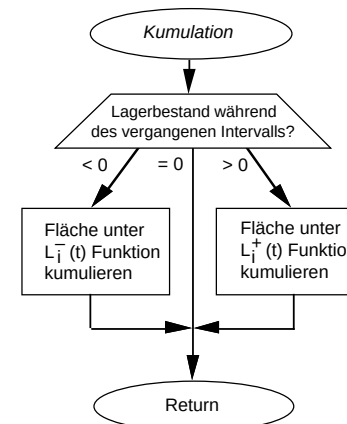
Prozedur	Zweck
<i>Initialisierung</i>	Einlesen der Eingabedaten, Vorbesetzung der Variablen (Statistische Zähler, Zustandsvariablen, Simulationsuhr, Startereignisse in Ereignisliste)
<i>Zeitführung</i>	Bestimmung des nächsten Ereignistyps und Weiterschalten der Simulationsuhr
<i>Lieferung</i>	Ereignisroutine für Ereignistyp <i>Lieferung</i>
<i>Kunde</i>	Ereignisroutine für Ereignistyp <i>Kundenauftrag</i>
<i>Simulationsende</i>	Ereignisroutine für Ereignistyp <i>Simulationsende</i> und Reportausgabe
<i>Bestellung</i>	Ereignisroutine für Ereignistyp <i>Bestellung</i> (Berechnung der Bestellmenge und Aufgeben der Bestellung)
<i>Kumulation</i>	Fortschreibung der Lagerbestände $L_i^+(t)$ und $L_i^-(t)$
<i>RandInt (N)</i>	Funktion zur Generierung von ganzzahligen Zufallszahlen zwischen 0 und N
<i>Gleich (A, B)</i>	Funktion zur Generierung von kontinuierlichen gleichverteilten Zufallszahlen im Intervall $[A, B]$
<i>Exponential (μ)</i>	Funktion zur Generierung von exponentialverteilten Zufallszahlen mit Mittelwert μ

Abb. 2-12: Ereignisroutine *Lieferung*Abb. 2-13: Ereignisroutine *Kunde*

In der Ereignisroutine *Lieferung* (Abb. 2-12) wird anfangs das Unterprogramm *Kumulation* zur Fortschreibung der Lagerbestände $L_i^+(t)$ bzw. $L_i^-(t)$ (Flächenbestimmung) aufgerufen, da sich zu diesem Zeitpunkt die Produktlagerbestände verändern. Dann werden die Zugänge den Lagerbeständen hinzugerechnet.

Die Ereignisroutine *Bestellung* (Abb. 2-14) vergleicht die tatsächlichen Lagerbestände L_i mit den minimalen Beständen s_i und bestellt bei Erreichung oder Unterschreitung von s_i Produkte im Umfang von $S_i - L_i$. Die Statistik der Bestellkosten wird fortgeschrieben und ein Ereignis *Lieferung* (Ankunft der bestellten Ware) vorgemerkt. Schließlich wird noch ein neues Ereignis desselben Typs *Bestellung* für den nächstmöglichen Bestellzeitpunkt (hier nach Ablauf eines Monats) generiert.

Die Prozedur *Kumulation*, die von verschiedenen Ereignisroutinen benutzt wird, dient der Flächenkumulation der Funktionen $L_i^+(t)$ und $L_i^-(t)$ für jedes Produkt i , jeweils wenn die Lagerbestände verändert werden (Abb. 2-15).

Abb. 2-14: Ereignisroutine *Bestellung*Abb. 2-15: Prozedur *Kumulation*

Simulationsergebnisse

Der Ergebnisreport für einen Simulationslauf ist im folgenden wiedergegeben:

Report des Modells Lagerhaltung	
Eingabedaten	
Anzahl Produkte:	3
Anfangslagerbestand je Produkt:	60 Stück
Lagerhaltungspolitik (s,S):	(60, 80)
Simulationsdauer:	120 Monate
Zwischenankunftszeit Nachfrage:	1 Tag
Bestellzeitpunkte:	alle 30 Tage
Lieferfrist (gleichverteilt):	15 bis 30 Tage
Kostenfaktoren der Lagerhaltung (für alle Produkte gleich):	
Kosten pro Bestellung:	32 GE
Bestellkosten pro Stück:	3 GE
Lagerhaltungskosten pro Stück und Monat:	1 GE
Fehlmengekosten pro Stück und Monat:	5 GE
Ausgabedaten	
PRODUKT 1:	
Bestellkosten (gesamt):	12291 GE
Lagerhaltungskosten (gesamt):	6963 GE
Fehlmengekosten (gesamt):	2453 GE
Gesamtkosten:	21707 GE
Kosten pro Monat:	181 GE
PRODUKT 2:	
Bestellkosten (gesamt):	15454 GE
Lagerhaltungskosten (gesamt):	6711 GE
Fehlmengekosten (gesamt):	6443 GE
Gesamtkosten:	28608 GE
Kosten pro Monat:	239 GE
PRODUKT 3:	
Bestellkosten (gesamt):	12839 GE
Lagerhaltungskosten (gesamt):	6816 GE
Fehlmengekosten (gesamt):	3777 GE
Gesamtkosten:	23532 GE
Kosten pro Monat:	196 GE

Die Simulationsergebnisse für alle vier zu untersuchenden Bestellpolitiken sind in Tabelle 2-1 zusammengefaßt. Die aufgeführten Beträge sind als Monatskosten in Geldeinheiten (GE) aufzufassen.

Tabelle 2-1: Simulationsergebnisse für die vier untersuchten Bestellpolitiken

Bestell-politik	Artikel	Gesamt-kosten	Bestell-kosten	Lager-kosten	Fehlmenge-kosten
(20, 60)	1	218	100	56	62
	2	277	123	53	101
	3	243	104	60	79
	alle	738	327	169	242
(20, 100)	1	247	116	78	53
	2	322	157	71	94
	3	256	120	81	55
	alle	825	393	230	202
(40, 100)	1	212	95	92	25
	2	283	131	87	65
	3	240	114	89	37
	alle	735	340	268	127
(60, 80)	1	181	102	58	21
	2	239	129	56	54
	3	196	107	58	31
	alle	616	338	172	106

Auf der Basis der durchschnittlichen monatlichen Gesamtkosten als Zielkriterium erscheint die Lagerhaltungspolitik (60, 80) für alle drei Artikel am günstigsten.

Auch bei diesem Beispiel sollten die Simulationsergebnisse vorsichtig interpretiert werden, da die statistische Genauigkeit der Simulationsergebnisse nicht untersucht wurde (s. hierzu 4.4.3).

2.4.3 Simulation von ausfallanfälligen Systemen

Der Begriff des *ausfallanfälligen Systems* bezeichnet technische Systeme, die unter dem Aspekt der Zuverlässigkeit betrachtet werden. Fortschreitende Automation führte dazu, daß technische Systeme zunehmend komplexer und damit auch störanfälliger wurden. Die Notwendigkeit eines möglichst **störungsfreien Betriebs** unter kostengünstigen Bedingungen ließ das Interesse an Fragen der Zuverlässigkeit technischer Systeme stark ansteigen.

Diese Systeme werden als Mengen von einzelnen Komponenten modelliert, die jeweils eine spezifische Lebensdauer haben. Um einem Systemausfall vorzubeugen, werden in regelmäßig durchgeführten **Wartungen** prophylaktisch Bauteile ausgetauscht, die ihre voraussichtliche Lebensdauer nahezu erreicht haben. Fällt ein Bauteil

aus, muß repariert werden. Sowohl Wartungen als auch Reparaturen kosten Zeit und Geld (Stillstandskosten, Produktionsausfallkosten und Personalkosten der Techniker). Die typische Fragestellung bei solchen Modellen lautet, mit welcher *Wartungsstrategie* die Systemverfügbarkeit maximiert bzw. die Kosten minimiert werden können.

Modelle ausfallanfälliger Systeme unterscheiden sich sowohl von Bedienungs-/Wartemodellen als auch von Lagerhaltungsmodellen. Der Unterschied zu Lagerhaltungsmodellen liegt vor allem im Wesen der modellierten Systeme. Lagerhaltungsmodelle modellieren passive Speicher für Objekte. Ausfallanfällige Systeme sind dagegen aktive technische Systeme im Dauerbetrieb, der nur durch Komponentenausfälle unterbrochen wird. Im Unterschied zu beiden anderen Modelltypen kommen hier auch **keine Kunden oder Aufträge** vor, die Ressourcen in Anspruch nehmen.

Bedienungs-/Wartemodelle und Modelle ausfallanfälliger Systeme werden oft im gleichen Anwendungsbereich eingesetzt: zur Modellierung von Fertigungssystemen. Aber während in Bedienungs-/Wartemodellen die Auslastung der Systeme bzw. die Wartezeiten der die Systeme belegenden Aufträge analysiert werden sollen, interessiert bei Modellen ausfallanfälliger Systeme die Verfügbarkeit der Systeme, unabhängig von deren Inanspruchnahme durch Aufträge oder Kunden. Dementsprechend gibt es in reinen Modellen ausfallanfälliger Systeme auch keine Warteschlangen.

Man kann sich jedoch ohne weiteres vorstellen, ein Fertigungssystem als Kombination aus beiden Modelltypen darzustellen, es also gleichzeitig als Bedienungs-/Wartesystem und als ausfallanfälliges System zu betrachten. Auch eine Kombination mit Lagerhaltungsmodellen ist möglich. So kann z. B. die schon mehrfach zitierte Frachterflotte auch als ausfallanfälliges System mit mehreren Bauteilen von begrenzter Lebensdauer modelliert werden. Je nach prognostizierter Lebensdauer würden die Teile bei einem Aufenthalt der Frachter im Heimathafen ausgewechselt (Wartung). Bei Ausfällen von Bauteilen auf See käme es zu der erwähnten flugzeuggestützten Reparatur.

Unter bestimmten restriktiven Annahmen können ausfallanfällige Systeme auch mit **analytischen** Zuverlässigkeitsmodellen abgebildet werden. Die Lösungen, die mit relativ einfachen Algorithmen ermittelt werden, sind allerdings nur isoliert verwertbar und für die Praxis selten von Nutzen. Daher ist die Zuverlässigkeitsanalyse technischer Systeme ein weiteres wichtiges Anwendungsgebiet für die Simulation.

2.4.3.1 Systemstruktur

a) Objekte

Die einzigen modellierten Objekte sind die **Komponenten** oder Bauteile des ausfallanfälligen Systems. Das wichtigste Attribut der Komponenten ist ihre **stochastisch bestimmte Lebensdauer**. Außerdem ist jede Komponente durch eine bestimmte **Austauschzeit** bei regulären Wartungen, eine (längere) Austauschzeit bei Reparaturen sowie durch den erforderlichen **Arbeitsaufwand** und den **Stückpreis** charakterisiert.

b) Relationen

Im allgemeinen wird ein ausfallanfälliges System als unstrukturierte Menge von individuellen Komponenten modelliert. Angemessener ist es jedoch, technische Systeme hierarchisch zu strukturieren, d. h. Systeme selbst wieder als Komponenten übergeordneter Systeme zu betrachten. Die Komponenten eines ausfallanfälligen Systems stehen damit in einer "setzt-sich-zusammen-aus"-Relation.

Für die Modellierung ausfallanfälliger Systeme ist jedoch nicht von primärer Bedeutung, aus welchen Subkomponenten sich eine Komponente zusammensetzt, sondern unter welcher Bedingung sie als ausgefallen gilt. Dies ist i. d. R. bereits der Fall, wenn mindestens eine Subkomponente ausgefallen ist. Es gibt aber auch **redundant** ausgelegte Systeme mit gleichen, parallel angeordneten Komponenten. In solchen Anordnungen müssen alle Komponenten ausfallen, bevor es zu einem Gesamtausfall kommt. Selbstverständlich können **Parallelanordnung** und **Seriellanordnung** von Komponenten beliebig komplex geschachtelt sein. Das Modell eines ausfallanfälligen Systems muß also nicht nur die Kompositionsstruktur des Systems abbilden, sondern auch berücksichtigen, ob zusammengehörige Subkomponenten parallel oder seriell angeordnet sind, ob also ein Ausfall durch UND- oder ODER-Verknüpfung ermittelt werden muß.

c) Aktivitäten

Wartung: In industriellen Anlagen hängt die Verfügbarkeit oder Zuverlässigkeit des Systems entscheidend von der Wartung des Systems ab. Die Wartung kann unter zwei verschiedenen Zielsetzungen erfolgen:

1. *Maximale Verfügbarkeit* unter Einhaltung einer Obergrenze für die Wartungskosten.
2. *Minimale Wartungskosten* unter Sicherstellung einer bestimmten Verfügbarkeit.

Es gibt ferner zwei grundlegende Arten von Wartungsstrategien: Strategien, bei denen erst *nach* einem Ausfall Teile ersetzt werden und Strategien, bei denen ausfallgefährdete Teile *vorbeugend* ausgetauscht werden. Vorbeugende Strategien sind oft von Vorteil, weil so die Wahrscheinlichkeit von Systemausfällen verringert und die Reparaturzeit minimiert werden kann. Zu den gängigen Strategien vorbeugender Wartung gehören:

1. *Opportunistische Strategien:* Ausfälle oder Produktionspausen werden zum vorbeugenden Austausch von Anlagenteilen genutzt.
2. *Streng periodische Strategien:* Für alle Anlagenteile gibt es feste Zeitspannen, nach deren Ablauf sie ersetzt werden.
3. *Pseudoperiodische Strategien:* Wenn ein Teil vor einem geplanten Austausch ausfällt und erneuert wird, schiebt man den Zeitpunkt des vorbeugenden Austausches um eine Periode hinaus, ersetzt es also nicht gleich wieder beim nächsten Austauschzeitpunkt.

4. *Sequentielle Strategien*: Nach jeder Wartungsaktion wird der Zeitpunkt und der Umfang (betroffene Anlagenteile) der nächsten Wartungsaktion bestimmt.

Ausfall/Reparatur: Die Aktivitäten bei einer ausfallbedingten Reparatur ähneln denen einer Wartung, allerdings werden sie *zufällig* ausgelöst und nicht entsprechend einer Wartungsstrategie geplant. Bei der Modellbildung ist zu berücksichtigen, daß Reparaturen bei Ausfällen i. a. länger dauern und höhere Kosten verursachen als die entsprechenden Wartungsaktivitäten.

d) Untersuchte Leistungsgrößen

Neben der Verfügbarkeit des Gesamtsystems ist bei einem Modell eines ausfallanfälligen Systems die Frage nach den **Kosten einer Wartungsstrategie** wichtig. Wie bei den Lagerhaltungsmodellen gehen mehrere Faktoren in die Gesamtkosten ein:

- Stückkosten der ersetzten Teile
- Personalkosten der Techniker für Reparatur und Wartung
- Produktionsausfallkosten des Systems

2.4.3.2 Beispiel: Wartungsstrategien für eine technische Anlage

Systemspezifikation

Eine ausfallanfällige technische Anlage bestehe aus 6 verschiedenen Komponenten, die über eine unterschiedlich lange **mittlere Lebensdauer** verfügen. Die Lebensdauer sei unabhängig Erlang-verteilt (s. 4.1.3). Für jede Komponente gelte eine bestimmte **mittlere Austauschzeit** (normalverteilt) mit einer festen Anzahl eingesetzter Mechaniker sowie bestimmten **Stückkosten**. In der folgenden Tabelle sind diese Daten zusammengestellt.

Tabelle 2-2: Komponentendaten der ausfallanfälligen Anlage

Teil Nr.	Mittlere Lebensdauer [Stunden] Erlang-verteilt mit $k=2$	Mittlere Austauschzeit [Stunden] normalverteilt μ σ	Anzahl der eingesetzten Mechaniker	Stückkosten [GE]
1	800	10 1.0	2	300
2	2200	6 0.8	1	130
3	1700	5 1.0	1	68
4	2900	6 0.5	2	71
5	3600	11 1.0	2	57
6	4800	7 1.0	1	44

Der Stundenlohn für einen Mechaniker betrage 50 GE.

Mit einem ereignisorientierten Simulationsmodell soll eine **strenge periodische Wartungsstrategie** getestet werden. Wie bereits erwähnt, werden bei solchen Strategien nach einem festen Zeitabstand Δt jeweils bestimmte Teile (Komponenten) der Gesamtanlage ausgetauscht. Der günstigste Zeitabstand ist zu bestimmen.

Für jede Strategie muß festgelegt werden, wie die jeweils auszutauschenden Teile ermittelt werden. Hier gelte: Ein Teil ist auszutauschen, wenn der Erwartungswert seiner Lebensdauer (Erlang-verteilt) im beginnenden Zwischenwartungsintervall oder im ersten Viertel des daran anschließenden Intervalls liegt.

Wenn der **Zeitpunkt einer Wartungsaktion** erreicht ist, werden die entsprechenden Teile ausgetauscht und die anfallenden Kosten berechnet. Es entstehen Materialkosten (Stückkosten gemäß Tab. 2-2) sowie Kosten für Arbeitslöhne und Produktionsausfallkosten. Die Arbeitskosten für den Austausch eines Teils errechnen sich aus dem Produkt der Austauschzeit für das betreffende Teil (μ , σ^2)-normalverteilt (s. Tab. 2-2) mit der Zahl der benötigten Mechaniker und dem Stundensatz von 50 GE. Die Produktionsausfallkosten schlagen mit 2500 GE pro Stunde zu Buche (Produktionsausfallzeit = längste Austauschzeit eines Teils). Es ist zu beachten, daß für neue Teile die Lebensdauer neu erzeugt werden muß. Außerdem muß das bereits angesetzte Ereignis vom Typ *Teileausfall* für das nun entfernte Teil aus der Ereignisliste gestrichen werden.

Beim **Ausfall** eines Teils ist die Anlage außerplanmäßig zu reparieren. Dabei entstehen neben den Kosten für die Reparaturaktion zusätzliche Kosten, da sich die Austauschzeit für das ausgefallene Teil um 12 bis 18 Stunden (gleichverteilt) gegenüber einem wartungsbedingten Austausch verlängert (Produktionsausfallkosten).

Mit dem Modell sollen für eine Planungsperiode von 240 000 Stunden verschiedene spezielle Wartungsstrategien realisiert werden, die sich nur in den Wartungsintervallen unterscheiden (800, 1000, 1200 Stunden usw.). Für jede spezielle Strategie sollen jeweils für Wartungsaktionen und für Ausfälle getrennt die durchschnittliche Austauschzeit, die durchschnittlichen Austauschkosten, die Gesamt-Ausfallkosten sowie die Summe aller Kosten ausgegeben werden. Außerdem ist die Systemauslastung für jede Strategie zu bestimmen. Wenn eine Planungsperiode beendet ist, wird festgestellt, ob die getestete Strategie eine minimale Verfügbarkeit der Anlage von 80% sicherstellt. Ist dies nicht der Fall, wird die Strategie verworfen. Unter den Wartungsintervallen, die die erforderliche Verfügbarkeit sicherstellen, ist das optimale auszuwählen.

Modellentwurf

Es werden drei **Ereignistypen** definiert (in der Reihenfolge ihrer Priorität):

1. *Ausfall*: Generierung eines Komponentenausfalls und Stillstand der Anlage.
2. *Simulationsende*: Ende der Simulationszeit des Simulationslaufs und Berechnung der Simulationsergebnisse (Report).
3. *Wartung*: Ersetzen von Teilen entsprechend der Wartungsstrategie.

Dem Ereignis *Simulationsende* wird gegenüber *Wartung* deshalb eine höhere Priorität zuerkannt, weil ein vorbeugender Teileaustausch auf zukünftige Maschinenausfälle ausgerichtet ist, die im aktuellen Simulationslauf nicht mehr erfaßt werden. Würden die entsprechenden Wartungskosten noch verbucht, so hätte dies eine Verfälschung der Ergebnisse zur Folge. Der prinzipielle Ablauf in den Ereignisroutinen *Ausfall* und *Wartung* ist in den Flußdiagrammen auf der folgenden Seite dargestellt (Abb. 2-16 und 2-17).

Es ist zu beachten, daß nur die reine Betriebszeit der Anlage ohne Reparatur- und Wartungszeit simuliert wird, da **nur während des Betriebes der Anlage** Ausfälle auftreten können.

Simulationsergebnisse

Die Simulationsergebnisse eines exemplarischen Simulationslaufs für eine spezielle Strategie (Wartungsintervall 1000 Std.) bei einer Betriebszeit von 240 000 Stunden sind im folgenden Simulationsreport dargestellt.

Report des Modells für eine ausfallanfällige Anlage

Eingabedaten:

Simulationszeit:	24000	Stunden
Wartungsintervall:	1000	Stunden

Ausgabedaten:

Systemverfügbarkeit:	94,2	%
Mittlere Reparaturzeit:	23	Stunden
Mittlere Reparaturkosten:	59533	GE
Gesamte Reparaturkosten:	31909730	GE
Mittlere Wartungszeit:	10,3	Stunden
Mittlere Wartungskosten:	28202	GE
Gesamte Wartungskosten:	6768418	GE
Gesamtkosten:	38678148	GE

Es sei wieder darauf hingewiesen, daß die Simulationsergebnisse hier nur zur Demonstration dienen. Sie können nicht als statistisch gesichert angesehen werden, da ihre statistische Genauigkeit nicht untersucht wurde.

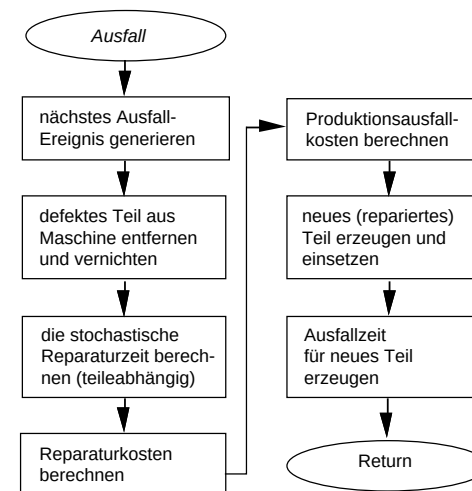


Abb. 2-16: Ereignisroutine für den Ausfall eines Maschinenteils

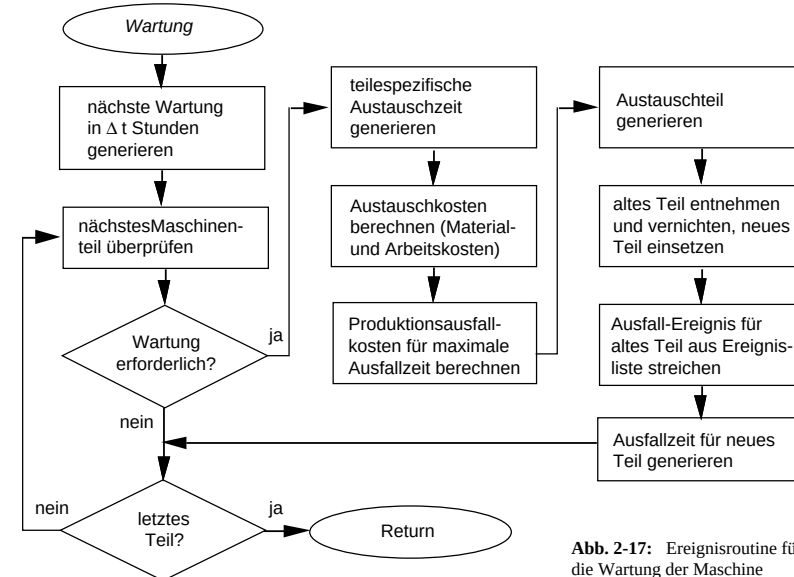


Abb. 2-17: Ereignisroutine für die Wartung der Maschine

2.5 Ereignisorientierte versus prozeßorientierte Sicht

In diesem Abschnitt sollen anhand eines einfachen Modells einer Fertigungsanlage (die als Bedienungs-/Wartesystem betrachtet wird) die prinzipiellen Unterschiede zwischen dem ereignisorientierten und dem prozeßorientierten "Weltbild" der Simulation verdeutlicht werden. Das Modell ist überschaubar und leicht verständlich, aber dennoch hinreichend komplex, um die grundlegenden Unterschiede dieser beiden wichtigsten konzeptuellen Sichtweisen der diskreten Simulation aufzuzeigen.

2.5.1 Beispielmodell eines Fertigungssystems

Die Analyse von Fertigungssystemen aus der Problemklasse der Bedienungs-/Wartesysteme stellt einen wichtigen Anwendungsbereich der diskreten Simulation mit ständig zunehmender praktischer Bedeutung dar. Generelles Ziel ist es, Engpässe in Produktionsprozessen aufzuspüren. Ein **Fertigungssystem** ist durch die folgenden Merkmale gekennzeichnet:

- **Betriebsmittelausstattung:** Die eingesetzten Maschinen mit ihren Leistungsdaten (entspricht den Ressourcen in Abschnitt 2.4.1.1).
- **Auftragsprofil:** Jede Auftragsart ist durch ein Auftragsprofil gekennzeichnet, bestehend aus einem Kennzeichen für die Auftragsart (Auftragskennzeichen), Betriebsmittelanforderung, Ankunftszeiten, Priorität usw.
- **Organisationsform:** Es muß dargestellt werden, in welcher Weise Aufträge auf Betriebsmittel verteilt werden, und in welcher Reihenfolge sie bearbeitet werden.

Die Merkmale Betriebsmittelausstattung, Auftragsprofil und Organisationsform bestimmen in enger Abhängigkeit voneinander das Leistungsverhalten des Fertigungssystems.

Zu den Leistungsgrößen, die mit dem Fertigungsmodell untersucht werden sollen, gehören z. B. die mittlere Warteschlangenlänge vor einer Maschine, die mittlere Bearbeitungszeit an einer Maschine (einschließlich Wartezeit), die mittlere Maschinenauslastung, der mittlere Durchsatz und die mittlere Verweilzeit im Fertigungssystem (vgl. auch 2.4.1.1).

Systemspezifikation

Das betrachtete Fertigungssystem bestehe aus fünf Maschinengruppen mit jeweils 7, 3, 8, 7 bzw. 3 identischen Maschinen zur Bearbeitung von Werkstücken. Es handelt sich somit um ein Wartenetz mit fünf Bedienstationen mit jeweils unterschiedlicher Zahl von Bedieneinheiten. Die Ankunft der Werkstücke folge einer Exponentialverteilung

mit $\mu = 0.10$ Stunden. Als Aufträge müssen drei verschiedene Werkstückarten (Typ 1-3), die mit unterschiedlichen Wahrscheinlichkeiten eintreffen ($p_1 = 0.3$, $p_2 = 0.5$, $p_3 = 0.2$), an einzelnen Maschinengruppen in unterschiedlicher Reihenfolge und Dauer bearbeitet werden. Die Bearbeitungsfolgen und -dauern der verschiedenen Auftragsarten sind wie folgt festgelegt:

Tabelle 2-3: Zusammenstellung der Auftragsdaten

Auftragsart	Wahrscheinlichkeit	Maschinenfolge	mittlere Bedienzeit in den einzelnen Maschinen [Std]
1	0.3	3, 1, 2, 5	0.50, 0.60, 0.85, 0.50
2	0.5	4, 1, 3	1.10, 0.80, 0.75
3	0.2	2, 5, 1, 4, 3	1.20, 0.25, 0.70, 0.90, 1.00

Wenn ein Auftrag bei einer bestimmten Maschinengruppe (Bedienstation) eintrifft und alle Maschinen dieser Gruppe (Bedieneinheiten) bereits aktiv vorfindet, wird er in eine unbegrenzte FIFO-Warteschlange eingereiht.

Abb. 2-18: Bearbeitungsreihenfolge für Auftragsart 1

Für die Bedienungszeit wird eine Erlang-Verteilung angenommen (s. 4.1.3). Mit dem Modell sollen 100 Acht-Stunden-Arbeitstage simuliert und die durchschnittlichen Wartezeiten für jede Auftragsart und für den gesamten Produktionsprozeß geschätzt werden. Außerdem sind für jede Maschinengruppe die durchschnittliche Warteschlangenlänge, die durchschnittliche Auslastung und die durchschnittliche Wartezeit abzuschätzen.

Anhand der Simulationsergebnisse könnte beispielsweise entschieden werden, ob bestimmte Maschinengruppen zahlenmäßig verstärkt werden müßten (*Engpaßanalyse*).

2.5.2 Ereignisorientierte Version

Gemäß der Systemspezifikation können die Aktivitäten und Ereignisse informal wie folgt beschrieben werden:

Nach der Erzeugung eines Auftrages durchläuft dieser, abhängig von der Auftragsart, zyklisch die folgenden Teilaktivitäten: Zunächst erfolgt die Ankunft in der ersten bzw. nächsten Maschinengruppe. Kann der Auftrag sofort bedient werden, wird die Bearbeitung des Teilauftrages über ein gewisses Zeitintervall vorgenommen, andernfalls

muß der Auftrag warten, bis eine Maschine frei wird. Bei Ende der Bearbeitung wird geprüft, ob weitere Maschinengruppen zu durchlaufen sind. Ist dies nicht der Fall, verläßt der Auftrag den Zyklus und damit das System (vgl. Abb. 2-19).

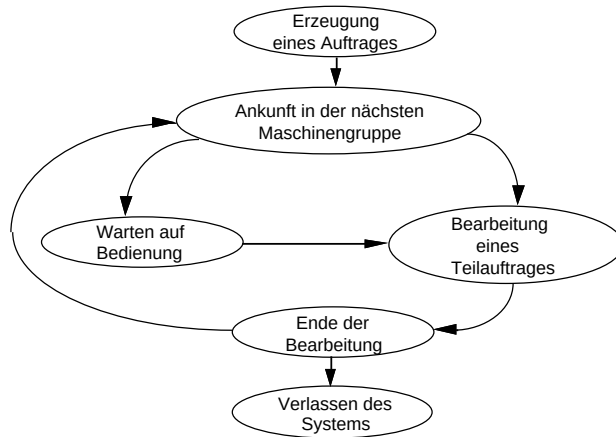


Abb. 2-19: Aktivitäten bzw. Ereignisse im Fertigungsmodell

Der oben beschriebene Ablauf läßt sich auf die folgenden **Ereignistypen** abbilden:

1. *Erzeugung eines neuen Auftrages*: Generieren und Vormerken eines Folgeauftrages, Festlegung der Auftragsart und Bearbeitungsreihenfolge des aktuellen Auftrages, Generieren und Vormerken der Ankunft an einer Maschinengruppe.
2. *Ankunft in einer Maschinengruppe*: Einreihung in die Warteschlange der Maschinengruppe oder sofortige Belegung einer Maschine (einschließlich Festlegung der Bedienzeit sowie Erzeugen und Vormerken des zugehörigen Bearbeitungsende-Ereignisses).
3. *Bearbeitungsende*: Ankunft in der nächsten Maschinengruppe vormerken oder System verlassen; danach Freigabe der Maschine oder Zuweisung eines Teilauftrages aus der Warteschlange zur weiteren Bearbeitung.

Die Abbruchbedingung ist hier nicht durch einen speziellen Ereignistyp *Simulationsende* realisiert, sondern als Abfrage auf Überschreitung einer vorgegebenen Simulationsdauer. Die Abläufe innerhalb der zugehörigen Ereignisroutinen sind als Flußdiagramme in den Abbildungen 2-20 bis 2-22 dargestellt.

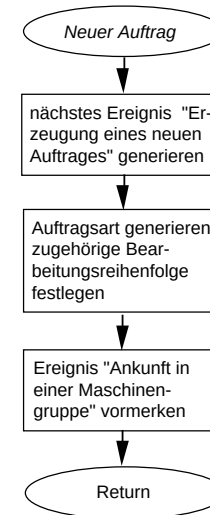


Abb. 2-20: Ereignisroutine zur Erzeugung eines neuen Auftrages

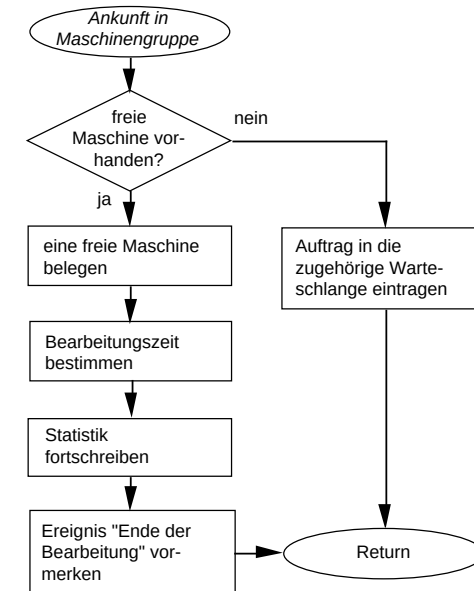
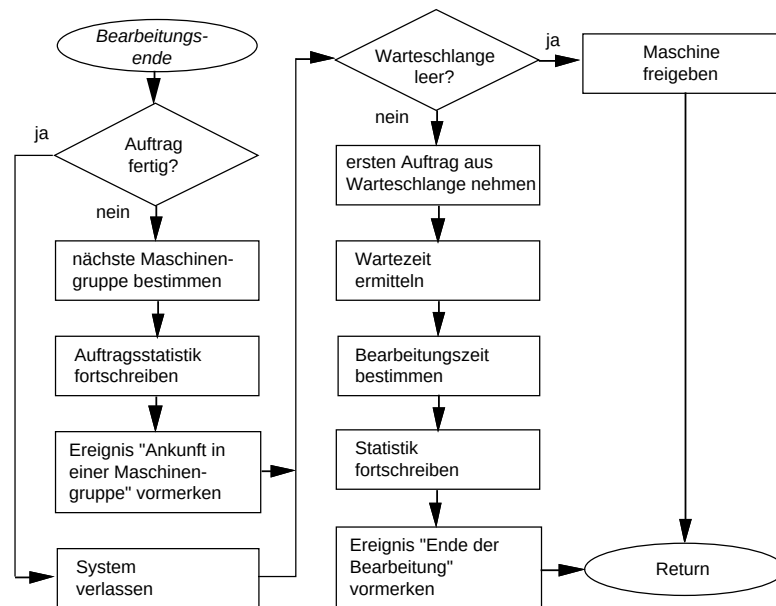


Abb. 2-21: Ereignisroutine zur Ankunft in einer Maschinengruppe

Abb. 2-22: Ereignisroutine *Bearbeitungsende*

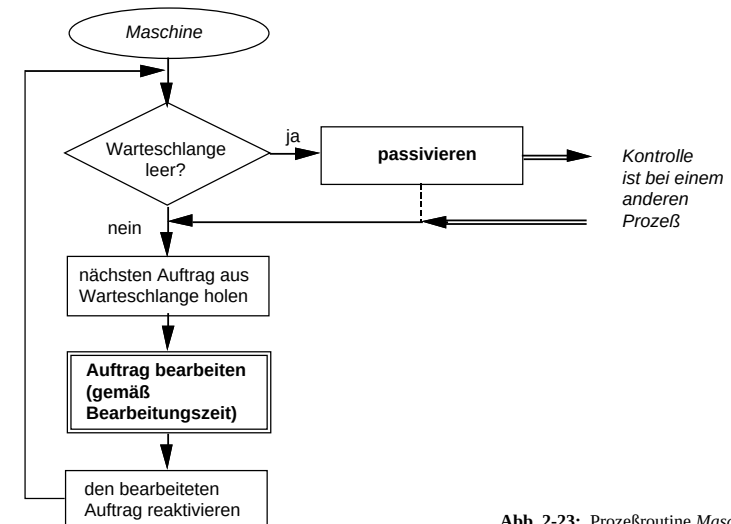
Bei der ereignisorientierten Sichtweise wird das dynamische Verhalten des Systems **als Folge von Ereignissen** beschrieben. Zu jedem Ereigniszeitpunkt wird im Modell eine entsprechende **Ereignisroutine** aktiviert, die die gewünschten Zustandsänderungen realisiert. Bezogen auf das Fertigungssystem treten die oben definierten Modellereignisse *Erzeugung eines neuen Auftrages*, *Ankunft in einer Maschinengruppe* und *Bearbeitungsende* auf. Die Struktur des Fertigungssystems (repräsentiert durch die Maschinenfolgen der Auftragsstypen) wird dabei völlig getrennt von seinem Verhalten betrachtet. Tätigkeiten, die in realen Fertigungssystemen zwischen den dargestellten Ereignissen stattfinden (z. B. die Bearbeitung eines Auftrags an einer Maschine), finden in dem Modell keine Berücksichtigung.

Das hier skizzierte ereignisorientierte Simulationsmodell wird in 3.3 in Modula-2 implementiert, wobei jedoch auf zuvor eingeführte Simulationsbausteine zurückgegriffen wird. Den zugehörigen Programmtext sowie exemplarische Simulationsergebnisse findet der Leser im Anhang B.2.

2.5.3 Prozeßorientierte Version

Im Gegensatz zur ereignisorientierten Sicht erfolgt hier für jede Systemkomponente eine Zusammenfassung ihrer Aktivitäten und Attribute zu einem **Prozeß**. Im oben beschriebenen Fertigungssystem umfaßt z. B. der Maschinenprozeß zum einen das Objekt Maschine und zum anderen die Aktionen, die diese an den ankommenden Aufträgen auszuführen hat. Aus Anwendersicht werden die Systemkomponenten durch diese Zusammenfassung anschaulicher. Darüber hinaus können sowohl die **aktiven** als auch die **passiven** Phasen einer Modellkomponente (z. B. *Bearbeitung* eines Auftrages durch eine aktive Maschine bzw. *Warten* einer passiven Maschine auf einen Auftrag) realitätsnah abgebildet werden.

In der prozeßorientierten Version des Fertigungsmodells werden sowohl die Maschinen als auch die Aufträge als zeitweilig aktive Modellkomponenten (und damit als Prozesse) abgebildet. Damit wird das besondere Kennzeichen des prozeßorientierten Ansatzes, die Modellierung der **Interaktion zwischen Prozessen**, deutlich erkennbar.

Abb. 2-23: Prozeßroutine *Maschine*

Gestrichelte Linien in den Flußdiagrammen der Prozeßroutinen (Abb. 2-23 und 2-24) stehen für passive Prozeßphasen unbestimmter Dauer. Der Prozeß passiviert sich, d. h. er gibt die Kontrolle an einen anderen Prozeß ab (nach außen führende Doppelpfeile)

und kann nach einer unbestimmten Zeitspanne von einem anderen Prozeß reaktiviert werden (nach innen führende Doppelpfeile).

Doppelt umrandete Kästchen stehen für inaktive Phasen mit fester Zeitdauer. Der Prozeß gibt die Kontrolle ab, hat sich selbst aber in der internen Ereignisliste bereits zur Reaktivierung vorgemerkt. (Ein vorgemerktter Prozeß ist weder aktiv noch passiv.) Beispielsweise ist die Aktivität der Bearbeitung auf diese Weise realisiert. Der Prozeß ist während dieser Verzögerung *programmtechnisch* inaktiv; die quasiparallele Abarbeitung der Prozesse (Koroutinenkonzept) macht es erforderlich, daß die Prozeßroutine in diesem Fall die Kontrolle abgibt. *Konzeptuell* ist der Prozeß in dieser Phase jedoch aktiv; die Verzögerung dient zur Darstellung einer zeitkonsumierenden Aktivität.

In der Prozeßroutine *Maschine* bleibt eine Maschine solange passiv, wie keine Aufträge in der Maschinengruppenwarteschlange auf Bearbeitung warten. Stehen Aufträge in Warteposition, so wird der erste der Warteschlange entnommen, gemäß der feststehenden Bearbeitungszeit bedient (zeitkonsumierende Aktivität, doppelt gerahmter Kasten) und zwecks Fortsetzung ihres Weges durch den Maschinenpark reaktiviert (Abb. 2-23).

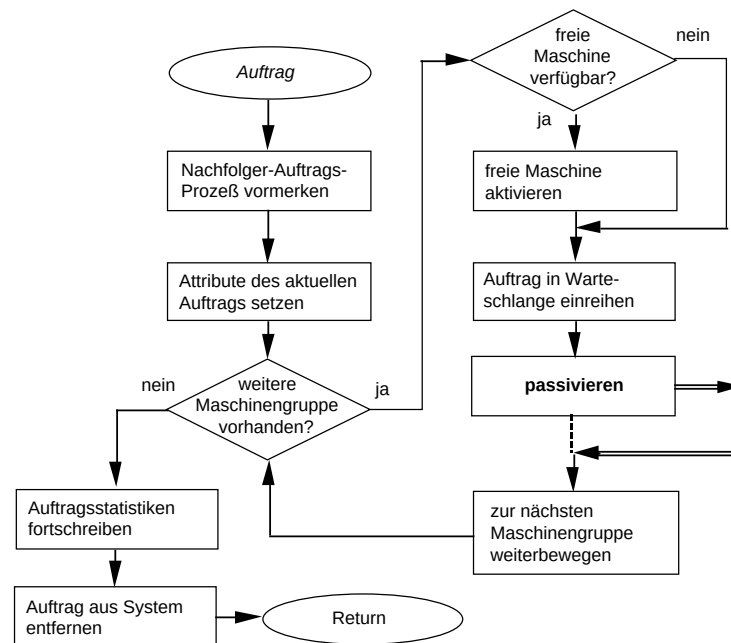


Abb. 2-24: Prozeßroutine Auftrag

In der Prozeßroutine *Auftrag* wandert der Auftrag nach dem Erzeugen und Vormerken eines Nachfolgeauftrages gemäß seiner Bearbeitungsreihenfolge von einer Maschinengruppe zur anderen, wobei er in jeder Gruppe zunächst ggf. eine untätige Maschine aktiviert, sich dann in die Auftragswarteschlange einreicht und auf Bearbeitung wartet, d. h. sich passiviert. Sofern eine freie Maschine verfügbar war, wird der vom aktivierten Prozeß *Maschine* zwecks Bearbeitung unmittelbar zu aktivierende Auftrag zum gleichen Zeitpunkt wieder aus der Warteschlange entnommen.

Aus Gründen der statistischen Erfassung der Aufträge ist es sinnvoll, in der beschriebenen Weise *alle* Aufträge in die Warteschlange einzufügen, also auch jene, die auf eine freie Maschine treffen. Die Wartezeit von Aufträgen, die sofort bedient werden können, beträgt Null. Bei dieser Vorgehensweise ist jedoch ein Problem zu beachten: Nach der Aktivierung einer freien Maschine durch den Auftragsprozeß behält dieser noch die Kontrolle, bis er sich in die Warteschlange eingereiht hat und sich passiviert. Bis zur Ausführung dieser Operation ist der Maschinenprozeß lediglich vorgemerkt. Erst nachdem sich der Auftragsprozeß passiviert hat, wird die Kontrolle dem nächsten vorgemerkten Prozeß übergeben. Um den vorgesehenen Ablauf zu garantieren, muß sichergestellt sein, daß hier kein anderer Prozeß als der vorgesehene Maschinenprozeß zum Zuge kommt. Es könnte insbesondere einen weiteren Auftragsprozeß geben, der zufällig zum gleichen Zeitpunkt vorgemerkt ist, nun aktiv wird und die gleiche Maschine zu belegen versucht wie der erste Auftragsprozeß. Um solche Probleme auszuschließen, stellen Simulationssprachen und -pakete des prozeßorientierten Ansatzes entsprechende Konstrukte zur Verfügung (in DESMO z. B. *ScheduleAfter*, s. Anhang C und 7.12.3).

Ein wichtiger **Vorteil** des prozeßorientierten Ansatzes gegenüber dem ereignisorientierten besteht darin, daß die insbesondere bei komplexeren Interaktionen unübersichtliche und fehleranfällige Auflösung zusammengehöriger Abläufe auf verschiedene Ereignisroutinen entfällt.