

1

Einführung in die Modellierung und Analyse ereignisdiskreter Systeme

Dieses Kapitel beschreibt die Grundideen der ereignisdiskreten Systembetrachtung und zeigt anhand von Beispielen aus unterschiedlichen Anwendungsgebieten, unter welchen Bedingungen die ereignisdiskrete Sicht auf dynamische Systeme sinnvoll bzw. notwendig ist.

1.1 Ereignisdiskrete Systeme

Die technische Entwicklung führt auf immer stärker von Computern geprägte, gesteuerte technische Systeme. Diese Systeme sind durch eine asynchrone Arbeitsweise ihrer Teilsysteme geprägt, für deren Informationskopplungen diskrete Ereignisse eine dominante Rolle spielen. Durch diese Ereignisse löst eine Komponente eine bestimmte Aktivität in einer anderen Komponente aus, stoppt einen Vorgang oder beeinflusst dessen zukünftiges Verhalten. Ereignisse beschreiben dabei beispielsweise das Einschalten eines Systems, das Erreichen einer bestimmten Position durch einen Roboterarm, die Weiterleitung eines Datenpakets oder die Grenzwertüberschreitung einer Temperatur. Darüber hinaus sind viele Entscheidungsregeln diskreter Natur. Sie beschreiben beispielsweise Situationen, in denen Komponenten eines technischen Systems ein- oder ausgeschaltet oder Rechenprozesse gestartet oder gestoppt werden sollen.

Um das Verhalten derartiger Systeme zu verstehen und zu analysieren, muss man Modelle verwenden, die die asynchrone Arbeitsweise der Teilprozesse und den Informationsaustausch

durch Ereignisse in den Mittelpunkt stellen. Diese Modelle, die die Grundlage der Theorie ereignisdiskreter Systeme bilden, werden in diesem Lehrbuch behandelt.

Das Verhalten ereignisdiskreter Systeme ist durch Ereignisfolgen gekennzeichnet. Jedes Ereignis stellt eine plötzliche Änderung eines Eingangs-, Zustands- oder Ausgangssignals dar, wobei idealisierend angenommen wird, dass der Signalwechsel keine Zeit in Anspruch nimmt. Die Signale haben zwischen den abrupten Signalwechseln konstante Werte, die zu einer endlichen Menge diskreter Signalwerte gehören. Der Signalverlauf lässt sich deshalb durch die Folge der nacheinander angenommenen Signalwerte oder durch die Folge der Ereignisse darstellen, die diese Signalwechsel bewirken. Diese Folgen beschreiben beispielsweise die zeitliche Änderung der Anzahl von Datenpaketen in einer Warteschlange, des Montagezustands eines Werkstücks oder der Menge der bereits ausgeführten Rechenoperationen.

Relationen zwischen diskreten Signalwerten bilden die Grundform der in diesem Buch erläuterten Modelle. Sie lassen sich im Allgemeinen nicht durch analytische Ausdrücke darstellen, so dass man die Tupel zusammengehöriger Signalwerte explizit aufschreiben muss. Graphentheoretische Verfahren bilden deshalb eine Basis für die Analyse des Systemverhaltens, und die Wahrscheinlichkeitstheorie wird verwendet, um bei nichtdeterministischen Zustandsübergängen die Häufigkeit zu beschreiben, mit der das System die einzelnen Zustandsübergänge durchläuft.

Die hier verwendete Betrachtungsweise unterscheidet sich grundlegend von den in den Ingenieurwissenschaften seit langem eingesetzten Methoden, die die kontinuierlichen Verhaltensformen technischer und natürlicher Systeme in den Mittelpunkt stellen und Material-, Energie- und Informationsflüsse durch sich stetig ändernde, quantitative Größen wie Druck, Temperatur, Massenflüsse oder Datenraten beschreiben und auf Modelle in Form von Differentialgleichungen und algebraischen Gleichungen führen.

Der grundsätzliche Unterschied zwischen kontinuierlichen und ereignisdiskreten Systemen liegt im Wertebereich der Signale. Während kontinuierliche Signale einen reellwertigen Wertebereich mit unendlich vielen möglichen Signalwerten haben, besitzen diskrete Signale einen Wertebereich mit vielen einzelnen Werten. Wenn in diesem Buch von kontinuierlichen oder diskreten Signalen oder Systemen gesprochen wird, so beziehen sich diese Attribute stets auf den Wertebereich der Signale. Dies wird hier betont, weil man in der Informationstechnik diese Attribute häufig in Bezug zur Zeit verwendet und unter diskreten Signalen dann reellwertige Signale versteht, die zu diskreten Abtastzeitpunkten betrachtet werden (vgl. Abschn. 2.3.1).

Gründe für die ereignisdiskrete Betrachtungsweise. Ob ein System durch ein kontinuierliches oder ein ereignisdiskretes Modell beschrieben wird, hängt nicht nur vom Charakter der in dem System ablaufenden Prozesse, sondern auch vom Modellbildungsziel ab. Wie die später behandelten Beispiele zeigen werden, ist es in vielen Situationen zweckmäßig, von kontinuierlichen Signalverläufen zu abstrahieren und Signale diskret zu beschreiben. Dieser Situation entsprechend gibt es zwei Gründe für eine ereignisdiskrete Betrachtung dynamischer Systeme:

- *Die Signale haben aus physikalischen oder technischen Gründen diskrete Wertebereiche.*

In dieser Situation hat man bei der Modellierung keine andere Wahl als mit diskreten Signalen zu arbeiten. So springen die internen Signale informationsverarbeitender Systeme in Abhängigkeit von der eingelesenen Zeichenkette zwischen diskreten Werten hin und her

und erzeugen eine wertdiskrete Ausgabefolge. Die ereignisdiskrete Betrachtungsweise ist deshalb in der Informatik sowie in wichtigen Gebieten der Informationstechnik weit verbreitet.

Auch beim Schalten eines Getriebes oder eines Schalters gibt es keine Zwischenwerte und man beschreibt den aktuellen Zustand durch eine ganzzahlige Größe. Bei der Paketvermittlung im Internet wird der Netzzustand durch das Vorhandensein oder Nichtvorhandensein von Datenpaketen dargestellt. In digitalen Schaltungen wird aus Gründen der Robustheit nur mit zwei Signalpegeln gearbeitet und durch die Schaltungstechnik erreicht, dass die kontinuierlichen Übergänge zwischen diesen Signalpegeln in so kurzer Zeit ablaufen, dass sie für die Funktionsweise der Schaltung nicht maßgebend sind (vgl. Abb. 1.2). Wenn ein System über Schaltventile gesteuert und das Verhalten über Endlagenschalter beobachtet wird, sind für das Modell ebenfalls nur diskrete Signalwerte maßgebend, obwohl unterlagerte physikalische Vorgänge durchaus in anderen Zusammenhängen kontinuierlich modelliert werden könnten.

- *Diskrete Signale entstehen durch Abstraktion aus kontinuierlichen Signalen.*

In dieser Situation werden bei der Modellierung nur diskrete Beschreibungen genutzt, obwohl die Signale reellwertig sind. Durch die Abstraktion werden die möglichen Zustände des Systems auf eine endliche Menge beschränkt. Damit verbunden ist die Darstellung kontinuierlicher Signaländerungen als Schalten des Signals von einem diskreten Wert zum nächsten, wobei sich der Nachfolgewert des Signals ohne Zeitverzögerung nach dem alten Signalwert einstellt.

Der Grund für diese Abstraktion liegt im Modellierungsziel, das seinerseits durch die betrachtete Analyse- oder Entwurfsaufgabe vorgegeben ist. So wird bei der Steuerung des Multitasking von Rechnern oder der Steuerung von Fertigungssystemen der Zustand des Gesamtprozesses durch die Angabe der bereits abgearbeiteten und der noch zu erledigenden Teilaufgaben beschrieben, weil es bei der Analyse nicht auf Einzelheiten der Teilprozesse, sondern auf deren Zusammenwirken ankommt. Ähnlich beruht bei Batchprozessen der Verfahrenstechnik eine Rezeptsteuerung auf der diskreten Betrachtung der Teilprozesse, bei der beispielsweise der Füllstand in einem Reaktor als „voll“ oder „leer“ gekennzeichnet wird, weil die vielen kontinuierlichen Zwischenwerte für den Beginn bzw. das Ende der Teilprozesse unwesentlich sind. Bei Ressourcenzuteilungsproblemen, die in der Fertigungstechnik Roboter für bestimmte Tätigkeiten einteilen, stellen die Zuteilung und die Freigabe von Ressourcen derartige Ereignisse dar.

Die ereignisdiskrete Betrachtungsweise beruht folglich in vielen Anwendungen auf einer Abstraktion, mit der die Behandlung des Systems auf das für die Lösung einer Aufgabe Wesentliche konzentriert wird. Dabei wird die Bewegung eines Systems durch Ereignisfolgen bzw. Folgen von diskreten Signalwerten dargestellt. Um die im Modell auftretenden Ereignisse mit den kontinuierlichen und diskreten Signalen eines technischen Systems in Beziehung zu setzen, muss man den betrachteten Prozess um einen *Ereignisgenerator* erweitern (Abb. 1.1). In diesem Block werden alle Operationen zusammengefasst, die aus den technisch auftretenden wertkontinuierlichen oder diskreten Signalen diskrete Werte- bzw. Ereignisfolgen erzeugen. Dies geschieht im einfachsten Fall durch die Detektion einer Grenzwertüberschreitung mit Hilfe ei-

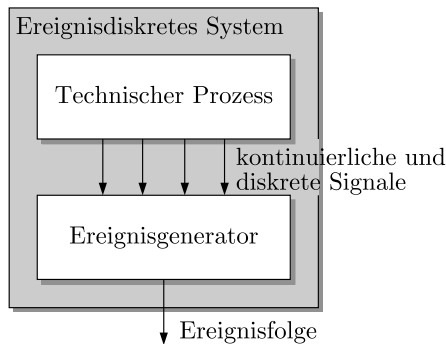


Abb. 1.1: Grundstruktur eines autonomen ereignisdiskreten Systems

nes Füllstandssensors oder durch das Signalisieren einer Position durch einen Endlagenschalter. Der Ereignisgenerator kann aber auch eine Methode zur Situationserkennung nutzen, um aus dem Verlauf eines oder mehrerer Signale das Auftreten eines Ereignisses zu erkennen. Schließlich können Ereignisse auch durch eine Uhr ausgelöst werden (Zeitereignisse). In allen Fällen macht der Ereignisgenerator in Abb. 1.1 die grundlegende Herangehensweise bei der Modellierung deutlich:

Bei einer ereignisdiskreten Modellierung wird das Verhalten des betrachteten dynamischen Systems auf die durch das System generierte Ereignisfolge reduziert und es wird angenommen, dass zwischen zwei aufeinander folgenden Ereignissen „nichts Interessantes“ passiert.

Beispiel 1.1 Ereignisdiskrete Arbeitsweise eines Getränkeautomaten

Die Kaffeezubereitung in einem Getränkeautomaten umfasst eine Reihe von Teilprozessen:

- Becher unter den Auslauf stellen
- Kaffeebohnen abmessen
- Kaffeebohnen mahlen
- Wasser zum Kochen bringen
- Kaffeepulver in einen Filter füllen
- Wasser durch den Kaffeefilter in den Becher laufen lassen.

Diese Prozesse laufen sequentiell oder parallel ab. Beispielsweise muss der Becher unter den Auslauf gestellt werden, bevor das kochende Wasser durch den Filter fließt. Andererseits kann man das Wasser zum Kochen bringen, während man gleichzeitig die Kaffeebohnen mahlt und in den Filter füllt.

Damit die genannten Teilprozesse im Getränkeautomaten in der gewünschten Reihenfolge und in der erforderlichen Dauer ablaufen, muss man sie zum richtigen Zeitpunkt starten und stoppen. Das Ein- und Ausschalten der entsprechenden Komponenten des Getränkeautomaten sind Ereignisse, die im Vergleich zu den beschriebenen Teilprozessen in vernachlässigbar kurzer Zeit ablaufen. Das Verhalten des Getränkeautomaten äußert sich deshalb als eine Folge dieser Ereignisse.

Der Getränkeautomat erfüllt seine Funktion, wenn die Ereignisse in der gewünschten Reihenfolge auftreten. Funktionsstörungen können deshalb daran erkannt werden, dass gewünschte Ereignisse nicht vorkommen, zusätzliche Ereignisse auftreten oder die Ereignisse die falsche Reihenfolge haben. Wenn man die Funktionsweise des Getränkeautomaten verstehen bzw. überwachen will, ist die Verhaltensbeschreibung durch eine Folge von Ereignissen zweckmäßig. Gleiches gilt für die Steuerung, die auf gemessene Ereignisse reagiert und Steuerereignisse vorgibt. Beispielsweise meldet die Kaffeemühle der Steuerung durch ein binäres Signal, dass die abgemessene Menge von Kaffeebohnen vollständig gemahlen ist. Nach diesem Ereignis schaltet die Steuerung die Heizung ein. □

1.2 Anwendungsgebiete der Theorie ereignisdiskreter System

Dieser Abschnitt zeigt anhand einer Reihe von Beispielen, in welchen Fachgebieten ereignisdiskrete Systeme verbreitet sind. Dabei wird offensichtlich, dass die ereignisdiskrete Betrachtungsweise dem Grundprinzip der Systemtheorie folgt, von den dynamischen Vorgängen soweit zu abstrahieren, dass das entstehende Modell nur die für die Lösung einer Aufgabe notwendigen Informationen erfasst.

1.2.1 Verarbeitung formaler und natürlicher Sprachen

Die Automatentheorie beschreibt die formalen Grundlagen der Informatik. Mit den in den Kapiteln 3 und 4 behandelten deterministischen und nichtdeterministischen Automaten können so fundamentale Fragen wie die nach der Berechenbarkeit von Funktionen oder nach der Komplexität von Algorithmen beantwortet werden.

Ein Grundproblem stellt die Frage, ob ein vorgegebenes Wort zu einer bestimmten Sprache gehört, wobei das Wort eine Zeichenkette und die Sprache eine Menge von Zeichenketten darstellt. Dieses Problem tritt beispielsweise bei der lexikalischen Analyse von Programmen auf, bei der ein Compiler nach Schlüsselwörtern wie „begin“ oder „end“ sucht. Ein Rechner soll das zu untersuchende Wort Zeichen für Zeichen einlesen, verarbeiten und am Ende der Zeichenkette die Antwort ausgeben.

Wie im Kapitel 3 beschrieben wird, kann man dieses Problem dadurch lösen, dass man den Rechner als Automaten darstellt, der seinen inneren Zustand nach dem Einlesen jedes Zeichens ändert und an dessen Endzustand man die Antwort auf die gestellte Frage ablesen kann. Befindet sich der Automat in einem festgelegten Endzustand, so sagt man, dass er die betreffende Zeichenkette akzeptiert. Die Menge aller von einem Automaten akzeptierten Zeichenketten nennt man die Sprache des Automaten.

Da man Rechner mit einer endlichen Anzahl von Komponenten realisieren will, darf seine Beschreibung nur endliche viele Zustände besitzen, was auf die Behandlung der angegebenen Probleme mit Hilfe von *endlichen Automaten* führt. Eine wichtige Erkenntnis der Automatentheorie ist die Charakterisierung aller Sprachen, die von geeignet gewählten Automaten akzeptiert werden (Abschn. 4.2). Obwohl diese sogenannten regulären Sprachen von einem endlichen Automaten akzeptiert werden, können sie unendlich viele Elemente enthalten.

Wendet man diese Betrachtungen auf Rechnerprogramme an, so fungiert der endliche Automat als ein Berechnungsmodell. Allerdings zeigt eine Analyse, dass man mit regulären Sprachen nur eine sehr eingeschränkte Klasse von Programmen darstellen kann. Um die Berechenbarkeit zu erweitern, muss man endliche Automaten zu Kellerautomaten und zur Turingmaschine erweitern (Abschn. 4.5). Damit hat man die allgemeinste Darstellung eines Berechnungsprozesses gefunden, denn nach dem allgemein akzeptierten Berechenbarkeitsbegriff gilt: Eine Funktion ist genau dann berechenbar, wenn sie mit einer Turingmaschine berechnet werden kann.

Neben der Berechenbarkeitstheorie beschäftigt sich das Gebiet der Computerlinguistik und Sprachtechnologie mit der Darstellung natürlicher Sprachen. In diesem Teilgebiet der Informatik werden Regeln zur Erzeugung und Erkennung natürlicher Sprachen aufgestellt, mit denen man beispielsweise Programme für die Rechtschreibkorrektur und automatische Silbentrennung von Textverarbeitungssystemen schreiben kann.

Mit den in diesem Buch behandelten regulären Sprachen können wichtige Eigenschaften natürlicher Sprachen nachgebildet werden. In der Computerlinguistik werden Erweiterungen vorgenommen, um den Aufbau und die *Grammatik* natürlicher Sprachen besser darstellen zu können. Dies bildet die Grundlage für die maschinelle Sprachverarbeitung mit dem Ziel der Informationsextraktion, des Textverstehens, der Inhaltsschließung elektronischer Dokumente, des maschinellen Übersetzens und dergleichen. Die dabei verwendeten Automaten mit Eingang und Ausgang (E/A-Automaten) lesen nicht nur den Text, sondern ordnen den erkannten Wörtern z. B. ihre grammatische Bedeutung zu. Sie werden in diesem Zusammenhang als *Transduktoren* bezeichnet.

1.2.2 Beschreibung eingebetteter Systeme

Eingebettete Systeme sind Rechner, die in einen technischen Kontext eingebunden sind. Die Rechenprozesse werden durch Ereignisse in der Umgebung des Rechners angestoßen oder beendet und die erhaltenen Ergebnisse wirken auf die Umgebung zurück.

Im täglichen Leben kommen wir ständig mit eingebetteten Systemen in Berührung, beginnend bei Mobiltelefonen und Spielcomputern bis zu Fahrkartenautomaten und Informationssystemen der Verkehrsbetriebe. Bei den genannten Beispielen umfassen die Interaktionen der in den Geräten vorhandenen Rechner mit der Umgebung auch die Kommunikation mit einem Bediener.

Bei der Verhaltensanalyse eingebetteter Systeme geht es beispielsweise um die Frage, ob es eine Bedienfolge gibt, bei der sich unterschiedliche Rechenprozesse gegenseitig verklemmen können, so dass das Gerät seine Funktion nicht mehr erfüllt. Dafür ist eine ereignisdiskrete Betrachtungsweise zweckmäßig, die sich auf die Interaktionen zwischen dem Rechner und seiner Umgebung konzentriert und diese durch Ereignisse beschreibt, die sich in einer Anfrage oder der Ausgabe der betreffenden Antwort äußern. Dieses Anwendungsgebiet ereignisdiskreter Modelle wird in der Literatur auch unter dem Stichwort *reaktive Systeme* behandelt.

Ein wichtiges Hilfsmittel zur Spezifikation und Modellierung eingebetteter Systeme ist die *Unified modeling language* (UML, vereinheitlichte Modellierungssprache), zu deren wichtigster Grundlage die im Kapitel 3 behandelten endlichen Automaten gehören. Diese Sprache ist heute ein internationaler Standard, auf dem viele Softwareentwicklungsumgebungen aufbauen.

1.2.3 Entwurf digitaler Schaltungen und Schaltkreise

Historisch betrachtet ist das Gebiet der digitalen Schaltungen das erste Gebiet, in dem die Theorie ereignisdiskreter Systeme technisch angewendet wurde. Die Veröffentlichungen von MEALY und MOORE, die um 1955 erschienen und diese Theorie entscheidend prägten, beschrieben eine Methode zur Spezifikation digitaler Schaltungen, mit der man die zu entwerfende Schaltung auch analysieren konnte. Zur damaligen Zeit war das wichtigste Analyseziel die Minimierung der zur Realisierung der Schaltung notwendigen Anzahl von Bauelementen, was auf eine Minimierung der Zustände der die Schaltung beschreibenden Automaten hinausläuft. Das im Abschn. 3.6.5 erläuterte Minimierungsverfahren für deterministische Automaten entstand in diesem Zusammenhang.

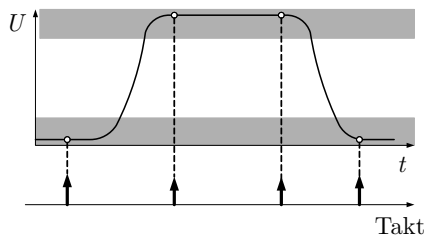


Abb. 1.2: Spannungsverlauf bei einer digitalen Schaltung

Im Unterschied zu anderen Anwendungsfeldern der Automatentheorie gibt es bei elektronischen Schaltungen einen Takt, was die Modellierung vereinfacht. Alle in der Schaltung ablaufenden Prozesse arbeiten synchron. Ein Taktgenerator gibt vor, zu welchem Zeitpunkt die am Eingang der Schaltung anliegenden Signale durch die Schaltung verarbeitet und die Ausgangssignale erzeugt werden. Aus dem im oberen Teil von Abb. 1.2 gezeigten kontinuierlichen Spannungsverlauf $U(t)$ wird zu den Taktzeitpunkten ein binäres Signal gebildet, das den Wert 1 hat, wenn sich die Spannung im oberen grauen Bereich befindet, und gleich null ist, wenn die Spannung im unteren Bereich liegt. Wenn die Taktfrequenz mit dem Zeitverhalten der Schaltung geeignet abgestimmt ist, liegt die Spannung bei jedem Taktzeitpunkt entweder im oberen oder im unteren Bereich und man kann von dem kontinuierlichen Übergang abstrahieren.

Da sämtliche in einer elektronischen Schaltung auftretenden Signale binär sind, wurde die Automatentheorie durch das Ziel vorangetrieben, Verfahren für eine automatische Synthese elektronischer Schaltungen zu erarbeiten, durch die eine logische Spezifikation der Funktionsweise der Schaltung in eine kombinatorische bzw. sequentielle Schaltungen überführt wird. Bei *kombinatorischen Schaltungen* hängen die Werte der Ausgangssignale nur von den zum selben Zeitpunkt an den Eingängen anliegenden Größen ab. Beispiele dafür sind die in elektronischen Schaltungen häufig vorkommenden logischen Glieder (ODER-Glied, UND-Glied usw.). Diese Schaltungen haben keine Speicher. Bei *sequentuellen Schaltungen* ist das Ergebnis von den Eingaben in Verbindung mit intern gespeicherten Zustandsgrößen der Schaltung abhängig. Die einfachsten sequentiellen Schaltungen sind Flipflops, die die beiden diskreten Zustände 0 und 1 besitzen und in Abhängigkeit von den binären Eingangssignalen zwischen diesen Zuständen

hin- und herspringen. Für die Beschreibung derartiger Schaltungen braucht man dynamische Modelle wie die im ersten Teil des Buches eingeführten Automaten (vgl. Aufg. 3.18 und 3.19).

Die ereignisdiskrete Beschreibung von Schaltungen und Schaltkreisen ist typisch für die oberen Betrachtungsebenen des Schaltkreisentwurfs (z. B. Gatterbeschreibungsebene), bei denen es um die logische Funktion der Bauelemente und Schaltungen geht. Teilschaltungen repräsentieren Logikblöcke, die im Modell unabhängig von ihrer schaltungstechnischen Realisierung als logische Funktionen vorkommen.

Ausgehend von der Darstellung digitaler Schaltungen durch Automaten wurden in den letzten drei Jahrzehnten unterschiedliche Beschreibungssprachen für elektronische Schaltkreise entwickelt, von denen beispielsweise VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) heute auch in anderen Gebieten wie mechatronischen Systemen eingesetzt wird. Ihre theoretische Grundlage ist die Beschreibung diskreter gekoppelter Systeme durch Automatenetze (Kap. 5).

1.2.4 Modellierung und Analyse von Fertigungssystemen

Wichtige Probleme der Projektierung großer Fertigungssysteme betreffen die logistische und zeitliche Planung (*Scheduling*) der Fertigung, wobei der Werkstücktransport bzw. die zeitliche Zuweisung von Ressourcen zu Fertigungsschritten im Mittelpunkt stehen. Zweckmäßigerweise erfolgt die Modellierung ereignisdiskret, wobei bei Montageprozessen jeder Montageschritt als ein Ereignis aufgefasst wird, das den Montagezustand des zu fertigenden Objektes verändert. Obwohl die Bewegung des dafür verwendeten Roboters oder der Werkzeugmaschine kontinuierlich ist, spielt bei der Fertigungsplanung nur ihre diskrete Darstellung eine Rolle.

Für die Analyse der Funktionsfähigkeit einer Fertigungszelle ist sogar eine *logische* Darstellung der auftretenden Ereignisfolge ausreichend, die sich nur auf die Ereignisnamen bezieht und keine Aussagen über die Ereigniszeitpunkte macht, denn es kommt zunächst nur darauf an, dass die Ereignisse in der richtigen Reihenfolge aktiviert werden. Soll auch der Durchsatz bestimmt und optimiert werden, so sind *zeitbewertete* Beschreibungsformen notwendig, die zusätzlich etwas über den zeitlichen Abstand der Ereignisse aussagen.

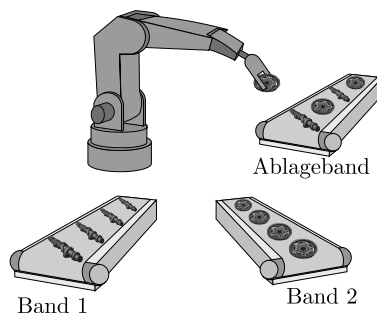


Abb. 1.3: Sortieren von Werkstücken durch einen Roboter

Beispiel 1.2 Handlungsplanung von Robotern

Abbildung 1.3 zeigt einen Teil einer Fertigungseinrichtung. Der Roboter soll die auf den Bändern 1 und 2 ankommenden Werkstücke des Typs A bzw. B auf das Ablageband transportieren und zwar so, dass sich die beiden Werkstückarten abwechseln.

Der Roboter kann Elementarbewegungen ausführen, zu denen das Öffnen und Schließen des Greifers und die Bewegung des Greifers zwischen den Bändern gehört. Diese Elementarbewegungen können als Ereignisse aufgefasst werden, weil die für diese Bewegungen notwendige Zeit für die Handlungsplanung nicht von Bedeutung ist. Vor bzw. nach diesen Ereignissen nimmt der Greifer einen der folgenden Zustände z ein:

z	Position des Roboterarms	Zustand des Greifers
1	über Band 1	geöffnet
2	über Band 1	Werkstück A gegriffen
3	über Band 2	geöffnet
4	über Band 2	Werkstück B gegriffen
5	über dem Ablageband	geöffnet
6	über dem Ablageband	Werkstück A oder B gegriffen

Jeder der sechs diskreten Zustände ist durch eine diskrete Position des Roboterarms und den Zustand des Greifers beschrieben.

Bei einer Handlungsplanung muss festgelegt werden, in welcher Reihenfolge die Elementarbewegungen ausgeführt werden, damit der Roboter eine vorgegebene Aufgabe erfüllt. Offensichtlich reicht dafür eine diskrete Betrachtungsweise aus, bei der die Elementarbewegungen die Ereignisse und die Anfangs- und Endpositionen der Elementarbewegungen die Zustände des Roboters repräsentieren. $\square$

1.2.5 Automatisierung diskreter Prozesse

Die Automatisierungstechnik beschäftigt sich mit der Überwachung und Steuerung technischer und nichttechnischer Systeme. Geräte und Anlagen sollen durch Automatisierungseinrichtungen so gesteuert werden, dass sie ihre Funktion erfüllen und Sicherheitsanforderungen genügen. Das Anwendungsgebiet der Automatisierungstechnik hat sich von klassischen Einsatzgebieten der Verfahrenstechnik und Energietechnik auf alle technischen Bereiche ausgeweitet und betrifft nicht nur größere Anlagen und Maschinen, sondern auch viele Konsumgüter. So läuft heute keine Waschmaschine und kein Heizkessel mehr ohne eine elektronische Steuerung und Kraftfahrzeuge sind mit bis zu vierzig vernetzten Steuergeräten ausgestattet.

Viele der zu steuernden Prozesse sind durch diskrete Zustandsübergänge und Ereignisse gekennzeichnet. So wird aus der Sicht der Steuereinrichtung, die den Fahrkorb eines Lifts entsprechend den Anforderungen der Nutzer in einer zweckmäßiger Reihenfolge zu den einzelnen Etagen führen soll, der Zustand des Fahrstuhls durch die Etage, in der sich der Fahrkorb befindet, bzw. die Bewegungsrichtung des Fahrkorbes gekennzeichnet. Batchprozesse in der verfahrenstechnischen Industrie sind in einzelne Stufen untergliedert, bei denen Stoffe erhitzt, abgefüllt, chemisch verändert oder mechanisch getrennt werden und deren erfolgreicher Abschluss durch ein Ereignis angezeigt wird. Viele Stellsignale, durch die die Steuereinrichtung

diese Batchprozesse realisiert, sind diskret, weil sie Motoren ein- oder ausschalten oder diskrete Arbeitspunktwechsel veranlassen.

Beispiel 1.3 Ereignisdiskrete Beschreibung eines Batchprozesses

Der in Abb. 1.4 dargestellte Ausschnitt aus einer verfahrenstechnischen Anlage mit den Behältern B_1 , B_2 und B_3 wird durch Öffnen und Schließen der Ventile V_1 bis V_6 gesteuert. Die mit L bzw. T gekennzeichneten Kreise sind Messstellen für Füllstände bzw. Temperaturen. In dieser Anlage soll ein Batchprozess ablaufen, wobei in der Ausgangssituation der Behälter B_3 leer ist, während die Behälter B_1 und B_2 gefüllt sind:

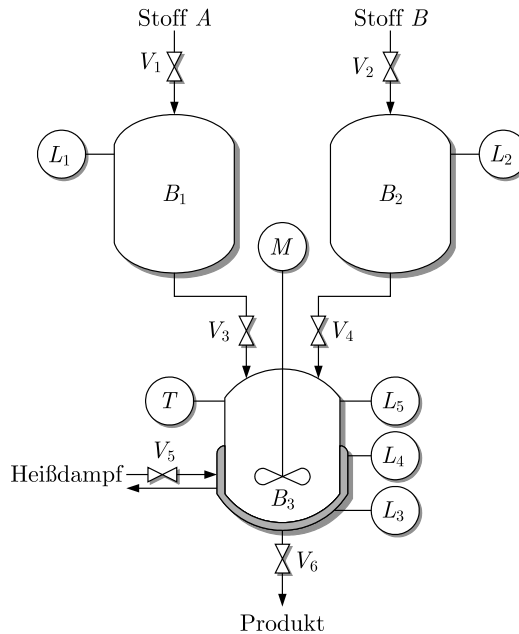


Abb. 1.4: Fließbild eines Batchprozesses

1. Es wird Flüssigkeit aus dem Behälter B_1 in den Behälter B_3 gefüllt, bis der Füllstand die Höhe des Sensors L_4 erreicht hat.
2. Es wird Flüssigkeit aus dem Behälter B_2 in den Behälter B_3 gefüllt, bis der Füllstand die Höhe des Sensors L_5 erreicht hat.
3. Wenn der Behälter B_3 gefüllt ist, wird der Rührer angeschaltet.
4. Wenn der Rührer angeschaltet ist, wird die Heizung angeschaltet und die Flüssigkeit im Behälter B_3 bis zu einer vorgegebenen Temperatur T_{soll} erhitzt.
5. Wenn die Flüssigkeit die Temperatur T_{soll} erreicht hat, werden Heizung und Rührer abgeschaltet und die Flüssigkeit wird aus dem Behälter B_3 abgezogen.
6. Während des Erhitzens der Flüssigkeit im Behälter B_3 werden die Behälter B_1 und B_2 bis zur Höhe der Sensoren L_1 und L_2 aufgefüllt.

Damit diese Teilprozesse in der richtigen Reihenfolge ablaufen, muss man den Prozess mit einer Automatisierungseinrichtung koppeln, die die Füllstände und Temperatur überwacht und die Ventile, die Heizung sowie den Motor des Rührers an- und ausschaltet. Die Steuerung reagiert auf diskrete Ereignisse, die das Erreichen einer vorgegebenen Füllhöhe der Behälter oder der Solltemperatur T_{soll} anzeigt. Diese Ereignisse führen dazu, dass die Steuereinrichtung den nächsten Stelleingriff ausgibt. Die Stellgrößen sind diskrete Signale, die die Ventile bzw. die Heizung ansteuern. Für den Entwurf der Steuerung ist es deshalb zweckmäßig, den Batchprozess ereignisdiskret zu betrachten und von den kontinuierlichen Vorgängen, die das Ansteigen bzw. Abfallen von Füllständen in den drei Behältern oder das Ansteigen der Temperatur kennzeichnen, zu abstrahieren.

Ereignisdiskrete Modelle werden für diese Automatisierungsaufgabe einerseits gebraucht, um den zu automatisierenden Prozess zu beschreiben. Das Modell muss beispielsweise aussagen, dass nach dem Öffnen des Ventils V_3 der Füllstand des zunächst leeren Behälters B_3 die Höhe des Sensors L_4 erreicht. Da von den kontinuierlichen Vorgängen abstrahiert wird, erscheint dieser Vorgang als ein diskretes Ereignis, durch das der Behälter B_3 vom Zustand „Füllstand befindet sich auf Höhe des Sensors L_3 “ in den Zustand „Füllstand befindet sich auf Höhe des Sensors L_4 “ übergeht. Andererseits werden diskrete Modell gebraucht, um die geforderte Funktionsweise des Prozesses bzw. für eine Sicherheitsanalyse verbotene Zustände und verbotene Zustandsübergänge zu spezifizieren.

Für den Entwurf der Steuerung reicht ein logisches Modell aus, das nur die Reihenfolge der Ereignisse erfasst (vgl. Beispiele 4.1 und 6.3). Es zeigt sich jedoch, dass dieses Modell nichtdeterministisch sein muss, weil beispielsweise nicht vorhergesagt werden kann, ob beim gleichzeitigen Füllen der Behälter B_1 und B_2 der Behälter B_1 oder der Behälter B_2 zuerst den Zustand „voll“ erreicht hat. Wenn man jedoch auch den Durchsatz durch den Batchprozess bestimmen will, so braucht man zusätzliche Informationen über die Dauer der beschriebenen Teilprozesse, die man mit zeitbewerteten Modellen erfassen kann (vgl. Kap. 9).

Verfahrenstechnische Batchprozesse sind ein gutes Beispiel dafür, dass die Entscheidung über eine kontinuierliche oder diskrete Modellierung vom Modellbildungsziel abhängt. Wenn man sich für die Automatisierungsaufgaben der Feldebene interessiert, also für die Geschwindigkeitsregelung von Rührern oder die Temperaturregelung von Reaktoren, so muss man die verfahrenstechnischen Komponenten als kontinuierliche Systeme betrachten und durch Differentialgleichungen beschreiben. Für die hier betrachtete Rezeptsteuerung, die man in der Hierarchie der Automatisierungsaufgaben der Prozessebene zuordnet, spielen jedoch die kontinuierlichen Vorgänge eine untergeordnete Rolle und können bei der Modellierung ignoriert werden. Derselbe Prozess, der für Regelungsaufgaben als kontinuierliches System betrachtet wird, wird hier ereignisdiskret beschrieben. $\square$

Diskrete Steuerungen. Die Grundstruktur einer diskreten Steuerung ist in Abb. 1.5 gezeigt. Das zu steuernde System erhält von der Steuereinrichtung zur Zeit k den diskreten Wert $v(k)$ der Stellsignale. In der ereignisdiskreten Betrachtung führt der zu steuernde Prozess daraufhin einen Zustandswechsel durch und gibt die Ausgangsgröße $w(k)$ aus. Die Steuereinrichtung bestimmt in Abhängigkeit von der Ausgabe $w(k)$ und vom Steuerungsziel den nächsten Wert der Stellsignale. Da es sich hier um rückgeführte Steuerungen handelt, spricht man auch bei Systemen mit diskreten Signalen bei der in der Abbildung gezeigte Struktur von einem *Regelkreis*.

Diskrete Modelle werden in diesem Zusammenhang nicht nur für den Entwurf der Steuereinrichtung, sondern auch für den Nachweis der korrekten Funktion des Regelkreises benötigt. Für sicherheitsrelevante Systeme braucht man dafür sogar einen formalen Nachweis, der als *Verifikation* bezeichnet wird (Absch. 3.5.3). Wichtige Eigenschaften sind die Einhaltung eines Sollverhaltens, die Widerspruchsfreiheit des Steuerungsalgorithmus oder die Lebendigkeit des

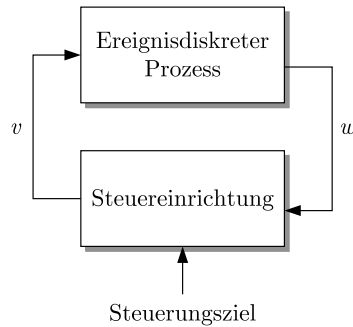


Abb. 1.5: Regelkreis

gesteuerten Systems, für die man das unaufhörliche Weiterschalten des ereignisdiskreten Modells unter allen Betriebsbedingungen nachweisen muss.

Modellbasierte Diagnose. Ein anderes Anwendungsgebiet der Theorie ereignisdiskreter Systeme in der Automatisierungstechnik ist die modellbasierte Fehlerdiagnose, mit der die fehlerfreie Arbeitsweise eines Prozesses bestätigt oder Fehler gefunden werden sollen. Der Diagnosealgorithmus prüft, ob das sich das aktuelle Systemverhalten, das sich in den Folgen von Eingabe- und Ausgabewerten äußert, mit dem Modellverhalten des fehlerfreien Systems übereinstimmt. Wenn dies nicht der Fall ist, kann darauf geschlossen werden, dass der Prozess fehlerhaft arbeitet.

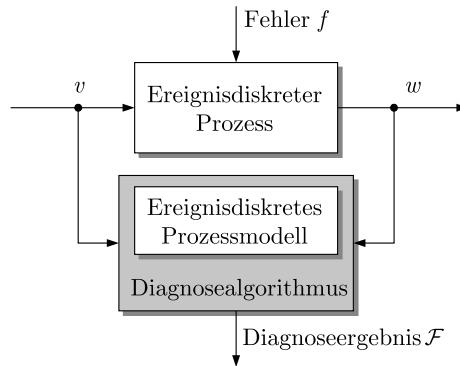


Abb. 1.6: Diagnose ereignisdiskreter Prozesse

Wie Abb. 1.6 zeigt, ist das ereignisdiskrete Modell ein zentraler Bestandteil des Diagnosealgorithmus. Wenn man nicht nur die Existenz eines Fehlers erkennen, sondern den Fehler sogar identifizieren will, braucht man außer dem Modell des fehlerfreien Systems auch Modelle, die das System unter der Wirkung von Fehlern beschreiben. Das Diagnoseergebnis ist i. Allg. eine

Menge $\mathcal{F}$ von Fehlerkandidaten, also von denjenigen Fehlern, für die der Prozess unter den Eingaben $v(k)$ die beobachtete Folge von Ausgaben $w(k)$ erzeugen kann.

1.2.6 Modellierung und Analyse von Kommunikations- und Rechnernetzen

Ein wichtiges Anwendungsgebiet ereignisdiskreter Modelle sind vernetzte Systeme, die aus einer großen Anzahl von verkoppelten Komponenten bestehen, wobei typischerweise jede Komponente nur mit einer vergleichsweise kleinen Anzahl von benachbarten Komponenten in direkter Beziehung steht. Derartige Systeme kommen beispielsweise als Informations-, Rechner- oder Verkehrsnetze vor.

Die strukturelle Beschreibung dieser Netze beruht auf gerichteten Graphen, deren Knoten die Verarbeitungseinheiten (Sende- und Empfangsstationen, Router, Rechner, Verkehrsknotenpunkte) und deren Kanten die Verbindungen dazwischen darstellen. Das Gesamtverhalten des Netzes wird entscheidend durch die begrenzte Verarbeitungsleistung der Knoten und die beschränkte Übertragungskapazität der Verbindungen bestimmt.

Um vernetzte Systeme analysieren zu können, muss man sich auf die wichtigsten Aspekte des Zusammenwirkens der Komponenten beschränken. Dabei spielt die ereignisdiskrete Sicht auf dynamische System eine entscheidende Rolle, denn sie abstrahiert von Einheiten der Funktionsweise der Verarbeitungseinheiten und der Kommunikation. So ist für das Verhalten eines Datennetzes am wichtigsten, welche Leitungen ausgelastet und welche frei sind, während der Verbindungsaufbau vernachlässigt werden kann.

Charakteristisch für das Verhalten netzförmiger Systeme ist die asynchrone Arbeitsweise der Komponenten, die auf Anforderungen an die Verarbeitung oder Kommunikation reagieren. Dies wird durch den vom Internet bekannten Begriff *asynchronous transfer mode* (ATM-Netz) ausgedrückt. Dem Internetprotokoll entsprechend werden die zu übertragenden Daten in Pakete zerlegt, die einzeln von Netzknoten zu Netzknoten übertragen werden. An jedem Netzknoten werden sie zunächst in eine Warteschlange eingeordnet und später entsprechend der im Kopf der Datenpakete angegebenen Adresse weitergeleitet. Wenn die Warteschlange keinen freien Platz hat, geht das Datenpaket verloren. Ein wichtiges Hilfsmittel für die schnelle und vollständige Übertragung der Daten ist deshalb ein Algorithmus, der die Rate, mit der die sendenden Knoten Datenpakete abschicken, der Netzbelastung anpasst.

Auch das Verhalten der Netzknoten kann ereignisdiskret beschrieben werden. Beispiele für die dort auftretenden Ereignisse sind die Ankunft einer Anfrage auf einem Server, das Zusammenstellen der Antwort und das Absenden der Antwort.

Die Modelle netzförmiger Systeme dienen nicht nur der Analyse des Netzverhaltens, sondern auch der Dimensionierung der Vermittlungsknoten und der Übertragungswege. Entsprechend dem diskreten Charakter der Netzbeschreibung und den diskreten Entscheidungsmöglichkeiten, eine Verbindung zwischen zwei Netzknoten zu realisieren oder wegzulassen, wird dieses Entwurfsproblem häufig durch diskrete Optimierungsverfahren gelöst.

Eine besonders enge Kopplung der Komponenten über Datenwege tritt bei *verteilten Systemen* auf. Dieser Begriff wird heute vor allem für Rechner verwendet, deren Software und Daten über mehrere Prozessoren verteilt sind und bei denen die Rechenprozesse in engem Informationsaustausch stehen. Die Aufgabe, den Ablauf der Rechenprozesse in ihrer gegenseitigen Verflechtung zu organisieren, führt zu einer ereignisdiskreten Betrachtungsweise, bei der nur

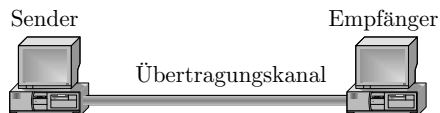
zwischen dem wartenden, arbeitenden und abgeschlossenen Prozess unterschieden wird und bei der die Kommunikation zwischen den Rechnern im Vordergrund steht.

Charakteristisch für das Verhalten netzförmiger Systeme ist die Tatsache, dass sich viele weitgehend unabhängige Prozesse gleichzeitig abspielen (nebenläufige Prozesse) und dass die auftretenden Ereignisse jeweils nur durch einen Teil der Komponenten beeinflusst werden. In den Zustand des Gesamtsystems gehen zu jedem Zeitpunkt viele Komponentenzustände ein, deren Änderungen sich weitgehend unabhängig voneinander vollziehen. Die Teilprozesse beeinflussen sich immer dann direkt, wenn sie gemeinsame Ressourcen nutzen.

Als Konsequenz dessen kann man die das Systemverhalten repräsentierende Ereignisfolge nicht genau vorhersagen. Wenn man beispielsweise die zeitliche Dauer der Teilprozesse nicht genau kennt, können die Ereignisse in unterschiedlicher Reihenfolge auftreten, je nachdem, welche Teilprozesse kürzer oder länger dauern. Das System ist nichtdeterministisch in dem Sinne, dass die für ein Modell zur Verfügung stehenden Informationen nicht ausreichen, um die in der Realität auftretenden Ereignisse in eine eindeutige Ordnung zu bringen. Dieser Nichtdeterminismus ist typisch für ereignisdiskrete Systeme und wird in diesem Buch noch an zahlreichen weiteren Beispielen erläutert.

Beispiel 1.4 Ereignisdiskrete Beschreibung einer Rechnerkommunikation

Die Übertragung von Daten zwischen zwei Rechnern über eine Netzwerkverbindung wird durch Protokolle geregelt. Diese legen fest, welche Daten zu welcher Zeit vom Sender und Empfänger gesendet bzw. empfangen werden. Auch wenn die Daten nur vom Sender zum Empfänger zu übertragen sind, ist eine bidirektionale Kopplung notwendig, damit der Sender anhand der erhaltenen bzw. ausgebliebenen Empfangsbestätigung erkennen kann, ob die Daten ordnungsgemäß beim Empfänger angekommen sind.



Zustände:

0 – Rahmen 0 wurde gesendet	E – leer	0 – wartet auf Rahmen 0
1 – Rahmen 1 wurde gesendet	0 – belegt mit Rahmen 0	1 – wartet auf Rahmen 1
	1 – belegt mit Rahmen 1	
	A – belegt mit Bestätigungsrahmen	

Abb. 1.7: Spezifikation eines Netzwerkprotokolls

Zur Illustration der ereignisdiskreten Arbeitsweise von Sender und Empfänger wird hier die Anwendungsschicht des Netzwerkprotokolls betrachtet, bei der die Daten und die Empfangsbestätigung in sogenannten Rahmen übertragen werden. Sender und Empfänger kommunizieren folgendermaßen:

1. Der Sender sendet den Rahmen 0 zum Empfänger.
2. Nach dem fehlerfreien Empfang von Rahmen 0 sendet der Empfänger den Bestätigungsrahmen zum Sender.
3. Der Sender sendet den Rahmen 1 zum Empfänger.

4. Nach dem fehlerfreien Empfang von Rahmen 1 sendet der Empfänger den Bestätigungsrahmen zum Sender.

Alle Rahmen können verloren gehen. In diesen Fällen erhält der Sender keine Bestätigung. Um das Ausbleiben der Bestätigung zu erkennen, läuft im Sender eine Uhr, die nach Ablauf einer vorgeschriebenen Zeit ein Ereignis erzeugt.

Bei der Kommunikation wird also nur zwischen gesendeten und nicht gesendeten bzw. erhaltenen und nicht erhaltenen Informationen unterschieden. Diese Betrachtungsweise ist ausreichend, um die ordnungsgemäße Übertragung von Daten zu sichern. Beispiel 3.4 auf S. 78 zeigt, wie Netzwerkprotokolle durch deterministische Automaten dargestellt werden können. □

1.2.7 Modellierung und Analyse von Wartesystemen

Wartesysteme bilden eine Klasse dynamischer Systeme, die in vielen Bereichen der Technik und des täglichen Lebens auftreten. Sie bestehen aus Warteräumen und Bedieneinrichtungen.

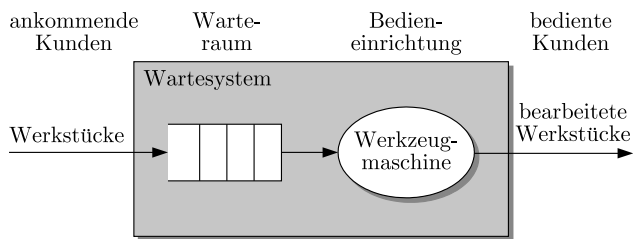


Abb. 1.8: Wartesystem mit einer Werkzeugmaschine

Warteschlangen entstehen durch Ressourcenmangel. Deshalb warten Druckaufträge in einer Druckerschlange auf den Ausdruck, Kunden an Bushaltestellen auf das nächste Fahrzeug oder Nachrichten und Daten auf das Freiwerden eines Übertragungskanals für die Übermittlung an den Adressaten. Platz für Warteschlangen wird deshalb in Rechnern in Form von Puffern vor Übertragungskanälen oder an Tankstellen vor den Tanksäulen geschaffen. Die genannten Prozesse haben trotz ihrer physikalischen Unterschiede viele systemdynamische Gemeinsamkeiten.

In der Grundform von Wartesystemen warten *Kunden* (Personen, Nachrichten, Aufträge, Waren, Autos) auf eine *Bedienung* (Ausführen eines Auftrags, Übermitteln von Daten, Freigabe einer Straßen usw.). In der in Abb. 1.8 gezeigten Symbolik wird der Warteraum durch „Fächer“ dargestellt, in denen die Kunden warten können, und die Bedieneinheit durch einen Kreis. Anordnungen, die aus einem oder mehreren Warteräumen und einer oder mehreren Bedieneinrichtungen bestehen, werden als *Wartesysteme*, *Warte-Bediensysteme* oder *Bediensysteme* bezeichnet.

Wartesysteme können vielfältige Eigenschaften aufweisen. In ihrer einfachsten Form ist ihr Verhalten dadurch gekennzeichnet, dass die Kunden einzeln ankommen, einzeln bedient werden

und die Bedienzeit nicht von der Warteschlangenlänge abhängt. Der Warteraum ist als FIFO-Speicher (*first-in first-out*) organisiert, so dass jeweils der Kunde mit der längsten Wartezeit als nächstes bedient wird. Die in diesem Buch behandelten Beispiele beziehen sich ausschließlich auf derartige Wartesysteme (Abschn. 9.4).

In der Praxis gibt es jedoch vielfältige Modifikationen dieser Grundform. So können Kunden, die einige Zeit vergebens auf eine Bedienung gewartet haben, die Warteschlange verlassen oder gegebenenfalls zwischen unterschiedlichen Warteschlangen wechseln. Die Kunden können einzeln oder in Pulks ankommen und möglicherweise in Gruppen bedient werden. Die Bedienzeit kann von der Warteschlangenlänge abhängen und auch von der Anzahl der bereits bedienten Kunden. Die Warteschlange kann unterschiedlich organisiert sein, so dass nicht wie bei den hier betrachteten Systemen stets derjenige Kunde als nächstes bedient wird, der am längsten gewartet hat. Schließlich kann die Bedienung aus unterschiedlichen Schritten bestehen, die nacheinander durchlaufen werden, ohne dass dazwischen weitere Warteräume angeordnet sind. Alle diese Erweiterungen dienen dazu, die Theorie den vielfältigen praktischen Gegebenheiten anzupassen. Wie bei der hier verwendeten Grundform von Wartesystemen ist die Beschreibung diskret und häufig durch Wahrscheinlichkeitsaussagen über die Anzahl der ankommenden Kunden und über die Ankunfts- und Bedienzeit geprägt.

Für Wartesysteme ist die ereignisdiskrete Betrachtungsweise zweckmäßig, weil sich die genannten Analyse- und Entwurfsprobleme auf das Vorhandensein einer bestimmten Anzahl von Kunden beziehen und die kontinuierliche Bewegung der Kunden in und aus der Warteschlange sowie der kontinuierliche Fortschritt der Bedienung von untergeordneter Bedeutung sind. Die Warteschlangen haben eine sehr einfache Beschreibung, denn sie sind Zähler mit einem Eingang und einem Ausgang. Allerdings reicht für die meisten Aufgaben eine logische Beschreibung der Zustandsübergänge nicht aus. Um Forderungen an den Durchsatz eines Wartesystems, die Verweilzeit der Kunden und die Reaktionszeit der Bediensysteme erfüllen zu können, muss man beim Entwurf dieser Systeme mit zeitbewerteten Modellen arbeiten. Da die Ankunfts- und Bedienprozesse nicht eindeutig vorhergesagt werden können, ist für sie eine stochastische Darstellung zweckmäßig. Aus diesen Gründen sind zeitbewertete Automaten eine geeignete Modellform (Kap. 9).

1.2.8 Zusammenfassung: Charakteristika ereignisdiskreter Systeme

Die in den vorangegangenen Abschnitten beschriebenen Beispiele haben gezeigt, dass es viele Prozesse gibt, deren Verhalten man in einer für die Modellierungsziele geeigneten Abstraktion als eine Folge von diskreten Zuständen bzw. Ereignissen beschreiben kann. Alle verwendeten Signale haben einen diskreten Wertebereich und ihr Wechsel zwischen diskreten Werten wird als Ereignisse interpretiert. Je nach dem Modellierungsziel interessiert man sich entweder nur für die Reihenfolge, in der die Ereignisse auftreten (logische Darstellung) oder muss auch die zeitlichen Abstände der Ereignisse erfassen (zeitbewertete Darstellung).

Bereits diese grundlegende Überlegung zeigt, dass man für ereignisdiskrete Systeme vollkommen andere Modelle benötigt als für kontinuierliche Systeme. Für sprungförmige Signaländerungen kann man einfach keine Differentialgleichungen aufschreiben!

Im Folgenden werden die charakteristischen **Eigenschaften ereignisdiskreter Systeme** zusammengestellt:

- Das Verhalten ereignisdiskreter Systeme ist durch Ereignisfolgen beschrieben, die bei autonomen Systemen spontane Zustandsänderungen beschreiben und bei gesteuerten Systemen von der Folge von Eingangsereignissen abhängig sind.
- Die Modelle unterscheiden sich bezüglich der Darstellung der Zeit: Logische Modelle sagen aus, *was* sich ereignet. Mit ihnen kann man die Reihenfolge bestimmen, in denen die Ereignisse auftreten. Zeitbewertete Modelle beschreiben, *was* sich *wann* ereignet. Sie geben zusätzlich Auskunft über die Zeitpunkte, an denen die Ereignisse auftreten. Welche Modellform verwendet werden muss, hängt von dem mit dem Modell zu lösenden Problem ab.
- Das Verhalten ereignisdiskreter Systeme wird durch Teilprozesse bestimmt, die typischerweise asynchron arbeiten und nur bei ausgewählten Ereignissen synchronisiert werden. Für die Modellbildung ist es wichtig zu identifizieren, welche Prozesse parallel und welche sequentiell ablaufen.
- Ereignisdiskrete Systeme können sich nichtdeterministisch verhalten. Der Grund dafür liegt in der mangelnden Kenntnis über das Systemverhalten, aufgrund dessen das Modell die von einem System erzeugte Ereignisfolge nicht eindeutig festlegen kann. Das System selbst befindet sich stets in genau einem Zustand und durchläuft genau eine Zustands- bzw. Ereignisfolge.

Die in diesem Buch behandelten Modelle können bei allen systemdynamischen Fragestellungen eingesetzt werden: Vorhersage und Analyse des Systemverhaltens, Systementwurf zur Erfüllung bestimmter Forderungen an das Systemverhalten. In der Informatik werden die Modelle auch als Berechnungsmodelle eingesetzt, die den Zusammenhang zwischen Spracheigenschaften und Rechenprozessen darstellen. Bei der Realisierung elektronischer Schaltungen dienen diese Modelle beispielsweise der Überprüfung der Vollständigkeit und Widerspruchsfreiheit der Spezifikation.

1.3 Überblick über die Modellformen und Analysemethoden

Modelle ereignisdiskreter Systeme. Die in den folgenden Kapiteln behandelten Modellformen sind in Abb. 1.9 zusammengestellt. Die Pfeile kennzeichnen die wichtigsten Beziehungen zwischen den Modellen.

- **Logische Modelle:** Als grundlegende Modellform ereignisdiskreter Systeme wird im Kapitel 3 der deterministische Automat eingeführt, der das logische Verhalten diskreter Systeme als eine Bewegung in einer endlichen Menge diskreter Zustände beschreibt. Im Kapitel 4 wird diese Modellform auf nichtdeterministische Automaten erweitert. Für die kompositionale Modellierung werden in den Kapiteln 5 und 6 Automatenetze bzw. Petrinetze eingeführt, die vor allem bei parallelen Prozessen zu einer Verringerung der Modellkomplexität beitragen.

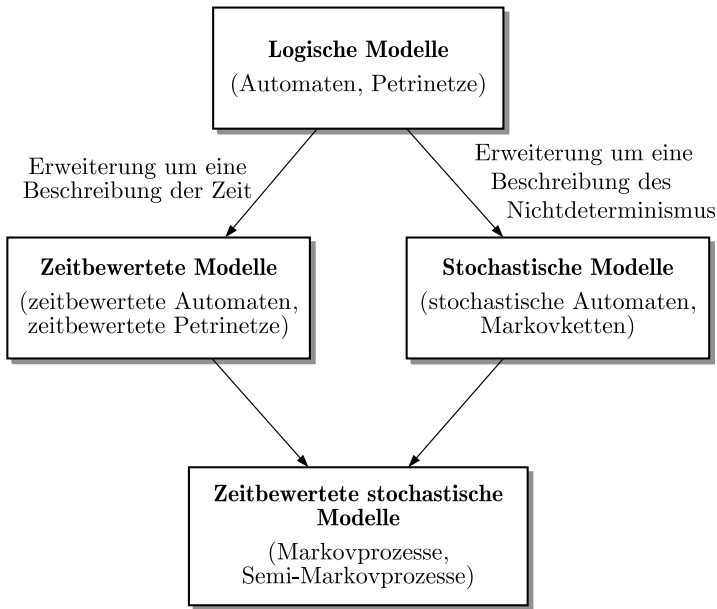


Abb. 1.9: Zusammenhang der in diesem Buch behandelten Modelle ereignisdiskreter Systeme

Die logischen Modelle lassen erkennen, *welche* Zustandsübergänge bzw. Ereignisse eintreten können. Sie ermöglichen die Analyse wichtiger Eigenschaften diskreter Systeme wie die Verklemmung bzw. Lebendigkeit, die Erreichbarkeit von erwünschten bzw. verbotenen Zuständen oder das Auftreten von verbotenen Zustandsfolgen. Logische Modelle sagen jedoch bei nicht-deterministischen Übergängen nichts darüber aus, welche Übergänge häufiger und welche seltener auftreten, und sie ignorieren die Zeitabstände zwischen den Zustandsübergängen. Um auch Aussagen darüber machen zu können, müssen logische Modelle zu stochastischen bzw. zeitbewerteten Modellen erweitert werden.

- **Stochastische Modelle:** Die Erweiterung logischer zu stochastischen Modellen hat das Ziel, die Aussagekraft der Modelle durch Hinzunahme von Informationen darüber zu verbessern, welche Zustände bzw. Zustandsübergänge häufiger und welche seltener auftreten. Dazu wird angenommen, dass das Auftreten der Ereignisse dem Zufall unterliegt und sich die Zustandsübergänge durch ihre Auftretenswahrscheinlichkeiten bewerten lassen. Mit dieser Erweiterung entstehen aus nichtdeterministischen Automaten stochastische Automaten (Markovketten, Kapitel 7).
- **Zeitbewertete Modelle:** Eine alternative Erweiterung logischer Modelle führt auf zeitbewertete Automaten und zeitbewertete Petrinetze, die Aussagen darüber machen, in welchem zeitlichen Abstand die Ereignisse auftreten. Dafür werden Petrinetze um Angaben darüber erweitert, wie lange die Marken in den Prästellen einer Transition liegen müssen, bevor die Transition schalten kann (Kap. 8). Nichtdeterministische Automaten werden um Anga-

ben über die Verweilzeit des betrachteten Systems in den einzelnen Zuständen erweitert (Kap. 9). Mit diesen Modellen kann man überprüfen, ob ein Prozess Zeitvorgaben einhält und welche Ereignisse zeitgleich bzw. zeitlich versetzt auftreten.

- **Zeitbewertete stochastische Modelle:** Die Erweiterungen logischer Modelle um eine wahrscheinlichkeitstheoretische Bewertung der Zustandsübergänge und um Aussagen über die Verweilzeit in den Zuständen können kombiniert werden, was auf die im Kapitel 10 behandelten Markovprozesse und Semi-Markovprozesse führt.

Mit den genannten Modellen behandelt das Buch die grundlegenden Beschreibungsformen ereignisdiskreter Systeme. Die Mehrzahl der in der Literatur vorgeschlagenen Modelle gehören entweder unmittelbar zu diesen Modellklassen oder sind Erweiterungen der behandelten Modelle und deshalb mit den hier vermittelten Kenntnissen verständlich. Ihre Vielfalt resultiert aus der Tatsache, dass Modelle diskreter Systeme eine große algorithmische Komplexität aufweisen und man deshalb daran interessiert ist, die spezifischen Verhaltensformen spezieller Systemklassen für eine Komplexitätsreduktion zu nutzen. Als Ergebnis dessen sind vielfältige Varianten der hier vorgestellten Modelle entwickelt worden, die auf bestimmte Anwendungsgebiete zugeschnitten sind. Kein Modell berücksichtigt alle Aspekte diskreter Verhaltensformen gleichermaßen.

Die im Folgenden vorgestellten Modelle zeigen auch, dass jede Erweiterung der grundlegenden Modellform „Automat“ die Aussagekraft der Modelle verbessert, dass dabei jedoch die Möglichkeit eingeschränkt wird, die Modelle mit effizienten Methoden zu analysieren. Für die hier eingeführten Modelle gibt es Analyseverfahren, die die Berechnung aller möglichen Zustandstrajektorien umgehen und aus strukturellen Eigenschaften des Automatengraphen bzw. des Petrinetzes wichtige Charakteristika der damit beschriebenen Modelle nachweisen. Für viele in der Literatur vorgeschlagene Modellerweiterungen kann man jedoch keine derartigen Verfahren entwickeln. Für diese Modelle kann man das Systemverhalten nur durch Simulationsuntersuchungen bewerten, wobei alle möglichen Verhaltensformen nacheinander durchgerechnet werden müssen. Zur Beschreibung diskreter Systeme sollte man also stets mit dem einfachsten Modell arbeiten, mit dem sich die wichtigen Eigenschaften des betreffenden Systems darstellen lassen.

Modellbildungsmethoden. Für die Anwendung der hier eingeführten Modelle ist nicht nur ihre Form, sondern auch der Weg wichtig, auf dem man für ein gegebenes System zu diesen Modellen kommt. Um die Strukturierung des Systems in verkoppelte Teilsysteme bei der Modellbildung nutzen, verwendet man die *komponentenorientierte Modellbildung* oder kompositionale Modellbildung, die für mehrere Modelltypen behandelt wird (Kap. 5 und 6).

Ein wichtiger Aspekt der komponentenorientierten Modellierung besteht in der Tatsache, dass die dabei entstehenden Modelle das asynchrone Verhalten der Teilprozesse geeignet wiedergeben können. Bei den Teilsystemmodellen nimmt man i. Allg. an, dass die am Eingang auftretenden Ereignisse eine sofortige Zustandsänderung und eine sofortige Ausgabe zur Folge haben, so dass Eingabe, Zustandswechsel und Ausgabe synchron auftreten. Bei der Verkoppelung der Teilsysteme wird dann jedoch deutlich, dass die synchronen Signalwechsel der Teilsysteme zu unterschiedlichen Zeitpunkten auftreten, die Teilsysteme also asynchron arbeiten.

Für die Analyse müssen nicht alle Teile des Modells mit demselben Detaillierungsgrad vorliegen, weil die Genauigkeit, mit der das System im Modell abgebildet sein muss, von der

konkreten Analyseaufgabe abhängt. Ein Hilfsmittel, mit dem diese Vorgehensweise umgesetzt werden kann, ist die *hierarchische Modellbildung*, bei der jede Systemkomponente in Modellen mit unterschiedlichem Detaillierungsgrad auftritt. Dieser Modellbildungsweg wird für Automaten und Petrinetze in den Abschnitten 3.7 und 6.4 behandelt.

Analyseverfahren. Ein Modell erfüllt erst dann seinen Zweck, wenn man mit ihm eine bestimmte Aufgabe lösen kann. In diesem Buch werden Modelle vor allem für die Analyse des Systemverhaltens eingesetzt, wobei unterschiedliche Analyseverfahren behandelt werden.

Ein wichtiges Hilfsmittel dafür ist die Systemdarstellung in Form von gerichteten Graphen, die die strukturellen Eigenschaften des Systems hervorheben. Graphentheoretische Methoden wie die Aufstellung des Erreichbarkeitsgraphen und die Zerlegung der Knotenmenge in Äquivalenzklassen stark zusammenhängender Knoten führen auf Aussagen darüber, welche Verhaltensformen möglich bzw. unmöglich sind. Dabei lässt sich nicht nur ein zyklisches Verhalten, das im Graphen durch geschlossene Pfade erkennbar ist, ablesen sondern auch eine Klassifikation der Zustände in transiente und rekurrente Zustände vornehmen, was direkte Aussagen bezüglich der Häufigkeit zulässt, mit der diese Zustände vom System angenommen werden (Abschn. 3.5 und 6.2).

Andererseits machen es algebraische Repräsentationen der Modelle möglich, wichtige Systemeigenschaften mit algebraischen Mitteln zu erkennen. So kann man die stationäre Verteilung der Zustände einer Markovkette aus der Matrixdarstellung des Modells ableiten und Invarianten für Petrinetze anhand deren algebraischer Darstellung definieren und analysieren.

Beide Hilfsmittel werden für alle eingeführten Modelle eingesetzt, wenn auch mit unterschiedlichem Gewicht.

Aufgabe 1.1 *Arbeitsweise eines Fahrkartenautomaten*

Betrachten Sie einen Fahrkartenautomaten, bei dem die Käufer zwischen verschiedenen Zielen, Fahrtrouten, Reisezeiten, Wagenklassen und Reservierungsmöglichkeiten wählen können. Welche Ereignisse bestimmen den Dialog zwischen dem Fahrkartenautomaten und dem Bediener und wie beeinflussen die Eingaben des Bedieners den weiteren Fortgang des Fahrkartenkaufs? □

Aufgabe 1.2* *Ereignisdiskrete Systeme?*

Das Verhalten der in der Tabelle zusammengestellten Systeme wird maßgeblich durch kontinuierliche Größen beschrieben. Nennen Sie Gründe dafür, dass man wichtige Eingangs- oder Ausgangsgrößen dieser Systeme dennoch als diskrete Signale behandelt.

Welche Ereignisse müssen Sie definieren, um das ereignisdiskrete Verhalten dieser Systeme beschreiben zu können? Welche dieser Ereignisse können direkt gemessen werden (und wie?), welche nicht?

Überlegen Sie sich weitere Beispiele! □

System	wichtige kontinuierliche Signale	diskrete Eingangs- oder Ausgangssignale
Stadtbeleuchtung	Tageslicht	Ein- und Ausschalten der Straßenbeleuchtung
Heizung	Vorlauftemperatur	Ein- und Ausschalten des Brenners
Scheibenwaschanlage eines Fahrzeugs	Verschmutzung	Ein- und Ausschalten der Scheibenwaschanlage
Straßenkreuzung	Verkehrsaufkommen an den Zufahrtsstraßen	Ampelschaltung
Batchprozess	Füllstand der Behälter	Alarmmeldungen, Ventile mit diskreten Stellungen
Rechner mit Multitasking	kontinuierlicher Fortgang von Rechenprozessen	Interruptsteuerung, Taskverwaltung
Flugverkehr	Position der Flugzeuge	Zuordnung der Flugzeuge zu Flugkorridoren
Vorlesung	Menge des zu vermittelnden Wissens	Vorlesungsplanung mit vorgegebenem Stundenrhythmus

Aufgabe 1.3* *Ampel rot – Motor aus*

Fahrerassistenzsysteme müssen diskrete Entscheidungen treffen bzw. dem Fahrer aggregierte Informationen übermitteln, obwohl viele der dabei zu berücksichtigenden Größen kontinuierlich sind. Diese Systeme haben deshalb die in Abb. 1.1 gezeigte Struktur, in der der Ereignisgenerator die kontinuierlichen Signale in diskrete überführt und dabei typischerweise mehrere Signale kombiniert.

Zeichnen Sie ein Fahrerassistenzsystem mit der in der Abbildung gezeigten Struktur, das den Motor bei Stillstandszeiten des Fahrzeugs an roten Ampeln oder im Stau abschaltet und den Motor wieder anschaltet, wenn der Fahrer weiterfahren will (Start-Stopp-Automatik). Schreiben Sie dafür zunächst die Bedingungen auf, unter denen der Motor ab- bzw. angeschaltet werden soll (z. B. abschalten, wenn der Motor warm genug ist, das Fahrzeug steht usw.). Kennzeichnen Sie in Ihrem Bild die Signale, die gemessen werden müssen, um diese Bedingungen prüfen zu können. Welche Signale werden zum Erkennen welcher Ereignisse benötigt? Ist das betrachtete Fahrerassistenzsystem ein statisches System, das zum Zeitpunkt t nur die jeweils gültigen Signalwerte verarbeitet, oder ein dynamisches System, das aus dem Verlauf der Signale bis zum Zeitpunkt t seine Entscheidung ableitet? ☐

Aufgabe 1.4* *Fahrgastinformationssystem*

Moderne Straßenbahnen und Busse sind mit Informationssystemen ausgestattet, durch die den Fahrgästen die nächste Haltestelle angekündigt wird. Früher musste der Fahrer rechtzeitig vor der Haltestelle durch einen Knopfdruck die Ansage auslösen; heute geschieht dies automatisch.

Betrachten Sie das Informationssystem als ein ereignisdiskretes System, das auf ein Eingangsereignis mit der Ausgabe des Namens der nächsten Haltestelle reagiert und stellen Sie es in der in Abb. 1.1 gezeigten Form dar. Welche Größen müssen Sie messen und wie können Sie aus diesen Größen das Eingangsereignis für das Informationssystem erzeugen? Beachten Sie dabei, dass bei einer Straßenbahn der Weg zwischen den Haltestellen gemessen werden kann, diese Messung aber mit Messfehlern behaftet

ist. Bei einem Bus ist der zwischen zwei Haltestellen zurückgelegte Weg nicht genau bekannt. Außerdem fahren Busse bei Haltestellen durch, wenn kein Fahrgast dort wartet und von den Fahrgästen nicht signalisiert wird, dass jemand aussteigen will. □

Aufgabe 1.5* *Handhabung eines Kartentelefons*

Beschreiben Sie den Ablauf des Telefonierens mit einem Kartentelefon, wobei Sie die Tätigkeiten des Benutzers als Eingaben und die vom Telefon ausgegebenen Informationen als Ausgaben auffassen. Ist es bei diesem Beispiel wichtig, die zeitlichen Abstände der Ereignisse in einem Modell zu erfassen? □

Aufgabe 1.6 *Ereignisdiskrete Beschreibung einer Eisenbahnverbindung*

Durch welche Ereignisse muss der Eisenbahnverkehr zwischen drei Bahnhöfen, die jeweils durch eine eingleisige Strecke verbunden sind, beschrieben werden, damit anhand dieser Beschreibung die Signale gestellt werden können? Welche zeitlichen Angaben sind notwendig, wenn das Modell darüber hinaus für die Fahrplangestaltung genutzt werden soll? □

Literaturhinweise

Eine der ersten Beiträge zur Theorie ereignisdiskreter Systeme stammt von C. E. SHANNON [94], der in einer 1938 erschienenen Arbeit zur Analyse von Relaisschaltungen die Grundlagen der Schaltalgebra legte. Betrachtet wurden dabei zunächst kombinatorische Schaltungen, die keine Speicher besitzen und deshalb statische Systeme darstellen. Die Arbeiten von G. H. MEALY [67] und E. F. MOORE [69] erweiterten die betrachteten Schaltungen um Speicher und führen den Automatenbegriff ein. Sie werden als die grundlegenden Arbeiten der Automatentheorie gewertet (vgl. auch [70]).

Die in der Automatisierungstechnik eingesetzten Methoden der Theorie ereignisdiskreter Systeme sind z. B. in [18, 59, 63, 83] beschrieben, die auch einen Vergleich zur Automatisierung kontinuierlicher Systeme enthalten. Die Aufgabe 1.5 ist dem Lehrbuch [63] entnommen. Andere Bücher wie [19] betrachten ereignisdiskrete Systeme als die Grundlage für Simulationsmodelle und -programme oder widmen sich Diagnoseaufgaben unter Verwendung von Automaten [12, 33]. Automaten bilden die Grundlage der theoretischen Informatik [39, 93, 107, 108] und werden in der Angewandten Informatik beispielsweise für die Lösung von Verifikationsaufgaben [8] eingesetzt.

Einige Beispiele dieses Buches für den Einsatz ereignisdiskreter Modellformen in der Sprachverarbeitung sind durch das Vorlesungsskript [34] angeregt worden.

Die Theorie der Warteschlangen (Warteschlangentheorie, Bedientheorie) entstand aus einer Analyse von Telefonnetzen zu Beginn des 20. Jahrhunderts. Der von A. K. ERLANG 1909 publizierte Aufsatz [22] markiert den Ausgangspunkt für eine langjährige Entwicklung, die nach 1950 in vielen theoretischen Arbeiten mündete und erst ab etwa 1980 zu praktisch anwendbaren Methoden und rechnergestützten Werkzeugen führte. Eine Einführung in diese Thematik ausgehend von stochastischen Prozessen gibt [10], eine tiefgründige Darstellung z. B. [31].

Eine Kurzbeschreibung der Sprache VHDL findet sich in [3].