Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Peter Scholz

Softwareentwicklung eingebetteter Systeme

Grundlagen, Modellierung, Qualitätssicherung

Mit 30 Abbildungen

Springer

Peter Scholz
Fachhochschule Landshut
Fachbereich Informatik
Am Lurzenhof 1
84036 Landshut
peter.scholz@fh-landshut.de

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über
http://dnb.ddb.de abrufbar.

ISSN 1439-5428
ISBN 3-540-23405-5 Springer Berlin Heidelberg New York

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutzgesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Text und Abbildungen wurden mit größter Sorgfalt erarbeitet. Verlag und Autor können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Satz: Druckfertige Daten der Autoren
Herstellung: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg
Gedruckt auf säurefreiem Papier 33/3142/YL - 5 4 3 2 1 0

Vorwort

„Man darf das Schiff nicht an einen einzigen Anker
und das Leben nicht an eine einzige Hoffnung binden!“

(Epiket, griechischer Philosoph, ca. 50 bis 138)

Eingebettete Systeme (engl. embedded systems) sind Computersysteme, die aus Hardware und Software bestehen und die in komplexe technische Umgebungen eingebettet sind. Solche Umgebungen können maschinelle Systeme wie etwa Kraftfahrzeuge, Flugzeuge, Fernsehgeräte, Waschmaschinen, Küchengeräte, Mobilfunktelefone u. a. sein, die der Interaktion eines menschlichen Benutzers bedürfen (z. B. Steer-by-Wire im Kraftfahrzeug) oder vollautomatisch agieren (z. B. Antiblockiersystem). Teilweise ist dieser Übergang auch fließend, so beispielsweise bei einem elektronischen Bremsassistenten.

Schon heute hat z. B. ein Mittelklassewagen Elektrik und Elektronik im Wert von rund 2.200 Euro an Bord. In zehn Jahren wird sich dieser Wert voraussichtlich auf ca. 4.200 Euro fast verdoppelt haben. Eingesetzt wird Informationstechnologie vor allem in den Bereichen Fahrwerksantrieb, Motormanagement, Sicherheit, Komfort, Emissionsreduzierung und Kommunikation/Entertainment bzw. Infotainment.

Eingebettete Systeme übernehmen komplexe Steuerungs-, Regelungs- und Datenverarbeitungsaufgaben für (bzw. in) diese(n) technischen Systeme(n). Ihre Funktionalität wird durch das Zusammenspiel von Spezialhardware, Standardprozessoren, Peripherie und Software realisiert.

Sie unterscheiden sich daher von anderen Systemen, wie beispielsweise Textverarbeitungs-, Buchhaltungs-, Internet- oder Warenwirtschafssystemen grundlegend und in mannigfaltiger Weise: Diese Systeme sind ausschließlich in Software realisiert, arbeiten

ggf. durch Interaktion mit einem menschlichen Benutzer vor einer Tastatur bestimmte Aufgaben ab, d. h. sie transformieren in einer ersten Näherung Eingaben in Ausgaben: Benutzereingaben werden in einem iterativen Prozess gelesen, bearbeitet und schließlich in Ausgaben transformiert. Bei ihrer Beschreibung liegt der Schwerpunkt auf den Transformationsprozessen bzw. -algorithmen selbst. Man spricht hier auch von „(rein) transformationellen" oder „interaktiven" Systemen.

Dem gegenüber stehen die reaktiven Systeme, die als Verallgemeinerung von eingebetteten Systemen fortwährend auf Eingaben ihrer technischen Umgebung bzw. eines technischen Prozesses reagieren und entsprechende, Kontext-abhängige Ausgaben erzeugen. Eingebettete, reaktive Systeme sind in eine – möglicherweise recht feindliche (Schmutz, Hitze, Kälte, schnelle Bewegung etc.) – Umgebung eingebettet. Bei ihnen liegt der Schwerpunkt der Beschreibung daher auf der Beschreibung der Interaktion zwischen System und Umgebung und damit auf dem Ein-/Ausgabeverhalten des Systems. Reaktive Systeme müssen zu jedem Zeitpunkt in einer nicht vom Computersystem selbst getriebenen Weise auf Sensordaten der Umgebung reagieren können und unterliegen hierbei oft sogenannten Echtzeitanforderungen.

Bei Echtzeitanforderungen unterscheidet man zwischen harten und weichen. „Harte" Echtzeitanforderungen müssen zur Erfüllung der Funktion eingehalten werden (Beispiel: Ein Airbag muss innerhalb von wenigen hundertstel Sekunden voll aufgeblasen werden, sonst verfehlt er seine Wirkung). „Weiche" Echtzeitanforderungen dagegen erhöhen den Komfort, man spricht hier gerade im Automobilbau auch von „Komfortelektronik" (Beispiele sind elektrische Fensterheber, Zentralverriegelung usw.). Wird das Thema „Embedded Systems" aus dem Blickwinkel der Ingenieursdisziplinen betrachtet, so wird hier naturgemäß gerne der Schwerpunkt auf Hardware-Gesichtspunkte (Mikrocontroller, Signalprozessoren, Sensoren, Aktoren, Analog-Digital-Wandler usw.) gelegt. Daraus könnte leicht der falsche Eindruck erwachsen, bei der Entwicklung eingebetteter Systeme handele es sich um eine reine Hardwareentwicklungsaufgabe. Dies ist aber mitnichten so. Vielmehr handelt es sich beim Entwurf eingebetteter Systeme um eine mindestens genauso wichtige Softwareentwurfsaufgabe. Gerade letztere ist Kern regen wissenschaftlichen Interesses (Rosenstiel, 2003), was es in überschaubarer Zukunft wohl auch noch bleiben wird. Ein aktueller Wegweiser für die Forschung und Lehre für das Software Engineering im Bereich eingebetteter Systeme findet sich in (Broy und Pree, 2003).

In diesem Buch wollen wir einen ersten Überblick über das Thema geben. Wir starten dabei nach einer ausführlichen Einleitung und Hinführung zum Thema mit der Diskussion von nebenläufigen Systemen. Danach widmen wir uns den Gebieten Echtzeit, Echtzeitsysteme und Echtzeitbetriebssysteme. Im Anschluss werden wir dann einen Überblick zur Entwicklung eingebetteter Systeme geben. An eingebettete Systeme werden oft Echtzeitanforderungen gestellt. Ein Echtzeitsystem ist ein System, bei dem der Zeitpunkt, zu dem Ausgaben erzeugt werden, bedeutend ist. Programme, die auf einer (fast) beliebigen Hardware ablaufen, die Grundfunktionen von Betriebssystemen erfüllen und Echtzeitverhalten aufweisen, nennt man Echtzeitbetriebssysteme. Wir beginnen daher mit einer Beschreibung von Echtzeitbetriebssystemen und widmen uns dann in den folgenden Kapiteln der Programmierung von eingebetteten Systemen, bevor wir auf ausgewählte Techniken zum Softwareentwurf für diese Systeme eingehen. Da es sich bei eingebetteten Systemen oft um sicherheitskritische Systeme handelt, deren Fehlfunktion ihre Umgebung massiv beeinträchtigen kann und letztendlich sogar zur Gefährdung von Menschenleben führen kann, ist die Qualität solcher Systeme von zentraler Bedeutung. Dieses Buch enthält daher ebenfalls eine überblicksartige Betrachtung des Themas Softwarequalität und schließt mit einer Zusammenfassung verschiedener für Embedded Systeme geeigneter Vorgehensmodelle.

Ein kompaktes Buch wie das vorliegende kann naturgemäß ein derart komplexes und umfangreiches Thema wie die Softwareentwicklung eingebetteter Systeme nicht auch nur annähernd erschöpfend in allen Details behandeln. Dieser Anspruch wird demnach selbstverständlich gar nicht erst erhoben. Vielmehr soll es einen Zugang zu diesem – gerade für die deutsche Softwareindustrie in den kommenden Jahren wohl sehr zentralen – Thema schaffen und „Lust auf mehr“ generieren. Ein besonderes aber nicht ausschließliches Augenmerk gilt dabei der automobilen Softwareentwicklung.

Bei der Auswahl der Inhalte habe ich mich dabei in erster Linie davon leiten lassen, wo ich vor allem aufgrund meines persönlichen Hintergrunds aus Lehre, anwendungsnaher Forschung und Praxiserfahrung aus Industrietätigkeiten Handlungsbedarf in den softwareerstellenden Unternehmen sehe. Viele dieser Inhalte konnte ich mit zahlreichen Teilnehmern aus meiner Weiterbildungsreihe „IT Update“ für Fach- und Führungskräfte persönlich und ausführlich besprechen. Insbesondere fanden dabei von mir angebotene Tagesseminare zu Themenkomplexen wie „Software Engineering“,

„Softwarequalität“ und „Software Engineering eingebetteter Systeme“ großen Zuspruch. Inhaltlich geht das Buch an jenen Stellen in die Tiefe, wo ich vor allem in den industriellen Forschungs- und Entwicklungsbereichen Entwicklungspotential sehe. Mit meinem Buch möchte ich daher den Praktiker genauso ansprechen wie Studenten der Informatik, Elektrotechnik oder Mechatronik im Hauptstudium, die erstmals Zugang zu diesem Thema suchen.

Eingebettete Systeme sind eine der schnellstwachsenden Branchen unter den Informatikanwendungen. In Zukunft darf sogar eher noch mit einer Zunahme dieses Ungleichgewichts gerechnet werden. Das Buch zeigt auch deutlich auf, wo in Zukunft interessante Aufgabengebiete und berufliche Chancen für Informatiker und Informationstechniker liegen werden. Die Darstellungsweise der Inhalte orientiert sich dabei gezielt an der Sprachwelt der Informatik.

Dieses Buch wurde ganz bewusst in Deutsch verfasst, ist aber stets bemüht, englische Fachbegriffe weitestgehend ebenfalls einzuführen. Da viele der tangierten Themenbereiche überwiegend auf einer englischsprachigen Terminologie basieren, wird gar nicht erst der Versuch unternommen, Anglizismen zu vermeiden.

Lesehinweis: Nach der Lektüre der Kapitel 1 bis 3 können die nachfolgenden Kapitel in beliebiger Reihenfolge gelesen werden.

Die Erstellung dieses Werkes wäre niemals ohne die tatkräftige Unterstützung Anderer möglich gewesen. Zunächst möchte ich meinen Studenten des Studienprojekts „Embedded Systems“ aus dem Jahre 2004 (den Herren Philipp Konradi, Matthias Ecker, Christian Könik, André Hofmann und Florian Kandlinger) am Fachbereich für Informatik der Fachhochschule Landshut danken, die mit ihrer Literaturrecherche wichtige flankierende Arbeiten geleistet haben. Mit den Ergebnissen aus ihrem, von mir betreuten Studienprojekt konnten sie den ersten Preis bei den „Audi IT Tagen“ im Herbst 2004 gewinnen, was mich letztendlich zum Verfassen dieses Buches beflügelt hat. Besonders bedanken möchte ich mich auch bei Frau Stephanie Hahn für die Übernahme des Erstlektorates. Weiterhin gilt mein spezieller Dank Frau Jutta Maria Fleschutz, Frau Gabi Fischer und Frau Dorothea Glaunsinger vom Springer-Verlag, die bei redaktionellen Fragen stets mit Rat und Tat zur Seite standen.

Ich widme dieses Buch in Dankbarkeit meinen Eltern.

Peter Scholz Februar 2005

Inhaltsverzeichnis

1 Einleitung

Kapitelübersicht

Das erste Kapitel enthält eine allgemeine Hinführung zum Thema. Diese beginnt mit einer kurzen Motivation und geht dann nach einer Klassifikation eingebetteter Systeme auf relevante Beispiele und betroffene Branchen sowie Anwendungen ein. Es folgt ein Überblick über die Struktur eingebetteter Systeme. Nach einer Reihe begrifflicher Definitionen schließt das Kapitel mit einer Diskussion zentraler Herausforderungen auf diesem Gebiet. Nach der Lektüre dieses Kapitels sollte der Leser die zentrale Rolle eingebetteter Systeme für die Informatik verstanden haben, sie von anderen Systemen unterscheiden können und die wichtigsten Anwendungen und Begriffe kennen, sowie die noch zu lösenden Herausforderungen auf diesem Gebiet einschätzen können.

1.1 Motivation

Eingebettete Systeme

Unter einem *eingebetteten System* verstehen wir ein in ein umgebendes technisches System eingebettetes und mit diesem in Wechselwirkung stehendes Computersystem. Es übernimmt dort komplexe Steuerungs-, Regelungs-, Überwachungs- und Datenverarbeitungsaufgaben und verleiht damit dem umgebenden System oft einen entscheidenden Wettbewerbsvorsprung. Solche Systeme sind heute auf breiter Front in alle Bereiche der Technik eingedrungen. Mehr als *neun von zehn* aller elektronischen Bauelemente sind in eingebetteten Systeme implementiert. In der Tat sind in modernen Personenkraftwagen der Oberklasse zwischen 70 und 80 integrierte und miteinander vernetzte Steuergeräte enthalten (Broy et al. 1998).

Reaktive Systeme

Reaktive Systeme nehmen heute in modernen Computersystemen eine bedeutende Rolle ein. Im Gegensatz zu rein transformationellen Systemen, also solchen, die lediglich Eingaben, die beim Systemstart vollständig zur Verfügung stehen, in Ausgaben, die erst bei der

Systemterminierung vollständig berechnet sind, verarbeiten, interagieren reaktive Systeme beständig mit ihrer Umgebung (Halbwachs, 1993), (Harel, Pnueli, 1985). Dabei wird die Interaktion des Systems mit seiner Umgebung weniger durch das System selbst getrieben, sondern von Ereignissen aus der Umgebung.

Reaktive Systeme finden ihre Anwendung in der Flugzeug-, Automobil- oder Telekommunikationselektronik. Bemerkenswerterweise sind bereits mehr Mikroprozessoren in reaktiven Systemen als in Personalcomputern eingebaut. So hat eine statistische Erhebung im Jahre 2002 ergeben, dass von 8,3 Milliarden weltweit produzierten Prozessoren 8,15 Milliarden in eingebetteten Systemen verbaut wurden, aber lediglich 150 Millionen in transformationellen Computersystemen (wie Personal Computer, Server, Mainframes usw.). Dies entspricht einem Verhältnis von 98:2 zugunsten der eingebetteten Systeme. In (Fränzle, 2002) ist dieser Zusammenhang aus Sicht eines eingebetteten Systems humoristisch und doch sehr treffend formuliert: „Ach wie gut, dass niemand daran denkt, dass mich ein Computer lenkt".

Eingebettete Systeme sind eine der schnellstwachsenden Branchen unter den Informatikanwendungen. In Zukunft darf sogar eher noch mit einer Zunahme dieses Ungleichgewichts gerechnet werden. Es zeigt auch deutlich auf, wo in Zukunft interessante Aufgabengebiete und berufliche Chancen für Informatiker und Informationstechniker liegen werden.

Die Hauptgründe für das große Interesse an eingebetteten Systemen sind Fortschritte in Schlüsseltechnologien wie Mikroelektronik und formalen Methoden zu ihrer exakten Beschreibung als Grundlage für Sicherheit und Zuverlässigkeit sowie die sich daraus ergebende Vielfalt von Anwendungen.

Aktuelle Trends

Aktuelle Trends beim Entwurf eingebetteter Systeme sind:

- Steigender Anteil des elektronischen Teilsystems, dabei steigender Anteil des digitalen Teilsystems sowie *steigender SW-Anteil,*
- Trend zu immer mehr Intelligenz und fortschreitender *Vernetzung.*
- *Entwurfskompromiss:* kostengünstige Standardkomponenten vs. schnelle Spezialhardware

1.2 Klassifikation, Charakteristika

Die heute verfügbaren Computersysteme können in drei unterschiedliche Klassen eingeteilt werden (Halbwachs, 1993), (Harel, Pnueli, 1985): (rein) transformationelle, interaktive und reaktive Systeme. Sie werden in erster Linie durch die Art und Weise unterschieden, wie sie Eingaben in Ausgaben transformieren (Scholz, 1998).

Transformationelle Systeme

Transformationelle Systeme transformieren nur solche Eingaben in Ausgaben, die zum Beginn der Systemverarbeitung vollständig vorliegen. Die Ausgaben sind dann nicht verfügbar, bevor die Verarbeitung terminiert. In solchen Systemen ist der Benutzer, oder allgemeiner, die Systemumgebung, nicht in der Lage, während der Verarbeitung mit dem System selbst zu interagieren und so Einfluss auf ihr Ergebnis zu nehmen.

Interaktive Systeme

Interaktive Systeme dagegen, also beispielsweise Betriebssysteme, erzeugen Ausgaben nicht nur erst dann, wenn sie terminieren, sondern sie interagieren und synchronisieren stetig mit ihrer Umgebung. Diese Interaktion wird durch das System selbst und nicht etwa durch seine Umgebung bestimmt: Wann immer das System neue Eingaben zur Fortführung weiterer Verarbeitungsschritte benötigt, wird die Umgebung bzw. der Benutzer hierzu aufgefordert (engl. to prompt) – das System synchronisiert sich auf diese Weise *proaktiv* mit seiner Umgebung. Wird diese Synchronisierung dagegen durch die Systemumgebung anstelle des Systems bestimmt, so nimmt das System selbst eine reaktive Rolle ein und wir sprechen von einem reaktiven System.

Ein bedeutender Unterschied zwischen den beiden Arten von Systemen ist der „Herr“ der Interaktion. Bei interaktiven Systemen ist es der Computer. Die „Anfragenden“ warten bis sie bedient werden, der Computer entscheidet wer, wann und wie behandelt wird. Nennenswerte Herausforderungen bei interaktiven Systemen sind die Vermeidung von Verklemmungen (engl. deadlocks), „Fairness“ und die Konsistenz verteilter Informationen. Bei reaktiven Systemen ist es dagegen die Umgebung, die vorschreibt was zu tun ist. Der Computer hat in der vorgegebenen Zeit auf Stimuli der Umgebung zu reagieren. Größte Belange sind die Sicherheit und die Rechtzeitigkeit (Berry, 1998).

Der Unterschied zwischen reaktiven und interaktiven Systemen hat großen Einfluss auf den Verhaltensdeterminismus in diesen Systemen. Interaktive Systeme werden größtenteils als nichtdeterministisch angesehen, denn der Computer trifft intern die Entscheidung ob und wann eine Anfrage beantwortet werden soll. Auch die Reak-

tion auf eine Sequenz der Anfragen muss nicht immer gleich sein. Im Gegensatz dazu ist das deterministische Verhalten ein fester Bestandteil eines reaktiven Systems. Die Reaktion muss eindeutig durch die Eingabesignale und evtl. deren zeitliche Reihenfolge definiert sein. Beispiel dafür ist die Steuerung eines Flugzeugs oder Autos.

Das Verhalten eines nichtdeterministischen Systems ist weitaus komplexer zu modellieren als das eines deterministischen. Die Charakteristik der beiden Systeme muss unbedingt bei der Entwicklung von geeigneten Techniken, Methoden und Werkzeugen berücksichtigt werden.

Reaktive Systeme

Die wichtigsten Eigenschaften *reaktiver Systeme* kann man wie folgt charakterisieren (Halbwachs, 1993): Sie arbeiten oftmals nebenläufig, müssen äußerst zuverlässig sein und dabei Zeitschranken einhalten. Sie können sowohl in Hardware wie auch in Software realisiert und auf einer komplexen, verteilten Systemplattform implementiert werden.

Die funktionale Korrektheit reaktiver Systeme spielt eine entscheidende Rolle bei der Entwicklung solcher Systeme. Nicht nur, dass sich die Markteinführung eines Produktes verzögern kann, wenn Systemfehler erst in einer späten Entwicklungsphase entdeckt werden, schlimmer noch sind kostspielige Rückrufaktionen, wenn Fehler erst dann bekannt werden, wenn das Produkt schon beim Endabnehmer angelangt ist. Schließlich können unentdeckte Fehler bei reaktiven Systemen in kritischen Anwendungen auch noch zu Konsequenzen von größerer Tragweite führen.

In aller Regel sind reaktive Systeme in eine komplexe, beispielsweise mechanische, chemische oder biologische Systemumgebung eingebettet. Es handelt sich dann um *eingebettete Systeme*. Mit Umgebung bezeichnen wir nicht nur die natürliche Umgebung in der das kontrollierende, reaktive System agiert, sondern auch den kontrollierten Teil des komplexen Gesamtsystems. Für solche reaktive, eingebettete Systeme ist in aller Regel nicht nur deren funktionale Korrektheit wichtig, sondern auch ihre Reaktionszeiten. Man spricht in diesem Zusammenhang von *Echtzeitsystemen*, vgl. Abbildung 1.1 (siehe auch Abschnitt 4).

Eingebettete Systeme lassen sich nach einer Reihe weiterer, unterschiedlicher Kriterien klassifizieren. Zum einen bietet sich eine Einordnung in die im folgenden Abschnitt 1.3 genannten Produktkategorien an. Zum anderen treten diese Systeme in unterschiedlichen technischen Ausprägungen auf. So lassen sich *kontinuierliche* von *diskreten* und *verteilte* von *monolithischen* Systemen unterscheiden. Enthält ein System sowohl kontinuierliches als auch

diskretes Verhalten, so spricht man von einem *hybriden* System (siehe Abschnitte 1.4 und 5.8).

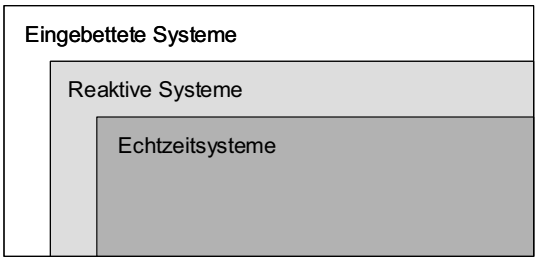

Abb. 1.1 Klassifikation eingebetteter Systeme

Sicherheitskritische Systeme

Eine weitere Zuordnungsmöglichkeit ergibt sich aufgrund der Sicherheitsrelevanz der Aufgaben, die ein eingebettetes System im Gesamtsystem übernimmt. Dies führt zur Einteilung in sicherheitskritische und nicht-sicherheitskritische eingebettete Systeme. Ein *sicherheitskritisches eingebettetes System* liegt vor, wenn von seiner korrekten Funktionsweise Menschenleben oder die Unversehrtheit von Einrichtungen abhängen. Während in der Konsumelektronik hauptsächlich nicht sicherheitskritische eingebettete Systeme zum Einsatz kommen, treten in der Avionik, Medizintechnik, insbesondere aber auch im Kraftfahrzeugbereich zunehmend sicherheitskritische eingebettete Systeme auf. Darüber hinaus lassen sich zeitkritische Systeme von nicht-zeitkritischen Systemen differenzieren. Bei letzteren spielt die Zeitspanne, innerhalb derer das System auf Eingabesignale reagiert, keine Rolle.

Damit zeichnet sich gerade die Automobiltechnik gegenüber den anderen später (siehe Abschnitt 1.3) genannten Produktkategorien durch zwei wesentliche Aspekte aus:

- Kraftfahrzeuge und damit die in ihnen (in Form von Steuergeräten) auftretenden eingebetteten Systeme bilden einen Massenmarkt.
- Eingebettete Systeme in Kraftfahrzeugen übernehmen zeit- und zunehmend auch sicherheitskritische Aufgaben.

Wir wollen uns daher bei unseren Betrachtungen im Folgenden vor allem auf eingebettete Systeme in der Automobiltechnik konzentrieren.

(Nicht-) Determinismus

Eingebettete Systeme müssen zu jedem Zeitpunkt ein *deterministisches*, also vorhersagbares Verhalten besitzen. Diese Eigenschaft gilt für ihre Realisierung, jedoch nicht notwendigerweise für ihre Anforderungs- oder Entwurfsspezifikation. In den Entwicklungsphasen der Anforderungsanalyse (engl. requirements analysis) bzw.

des Enwurfs (engl. design) kann die Spezifikation eines eingebetteten Systems durchaus Nichtdeterminismen aufweisen. Hier handelt es sich im Falle der Verhaltensspezifikation um das semantische Ergebnis gewünschter Unter- bzw. Überspezifikationen. Mit beiden Spezifikationsmitteln wird der Entwickler in die Lage versetzt, Teile des Systemverhaltens noch offen zu lassen, d. h. davon zu abstrahieren. Eine Konkretisierung der Spezifikation und damit Auflösung der Nichtdeterminismen muss zu einem späteren Zeitpunkt, spätestens jedoch zur Implementierung des Systems geschehen. Eine an das geschilderte Vorgehen angepasste Entwurfssmethode ist die sogenannte schrittweise Verfeinerung bzw. der inkrementelle Entwurf.

1.3 Anwendungen, Beispiele und Branchen

Anwendungsgebiete eingebetteter Systeme

Eingebettete Systeme sind heute wie bereits eingangs erwähnt und mit Zahlenmaterial unterlegt schon weit verbreitet. Sie sind insbesondere in folgenden Anwendungsgebieten zu finden:

- Telekommunikation (Vermittlungsanlagen, Mobiltelefone, Telefone, Faxgeräte etc.)
- Haushalt (Waschmaschine, Mikrowelle, Fernseher etc.)
- Geräte für Freizeit und Hobby
- Automobiltechnik (ABS, Wegfahrsperre, Navigationssysteme, Verkehrsleitsysteme etc.)
- Öffentlicher Verkehr (Fahrkartenautomat, etc.)
- Schienenverkehr (ICE, TGV)
- Luft- und Raumfahrttechnik, Avionik (Airbus-Reihe, Boeing 777, Militärsektor)
- Fertigungstechnik, Anlagenbau
- Steuerungs- und Regelungstechnik
- Medizintechnik
- Umwelttechnik
- Militärtechnik
- u. v. m.

Avionik und Automobilbau

Besonders interessante Anwendungsgebiete eingebetteter Systeme sind die *Avionik* (Flugzeugbau) und der *Automobilbau*. In einem modernen Fahrzeug sind heute oftmals mehr als 70 Mikroprozessoren oder elektronische Kontrolleinheiten integriert, die wir unter dem Begriff Steuergeräte zusammenfassen. Die Zahl der Prozessoren ist in den letzten Jahren sprunghaft gestiegen und wird auch in Zukunft noch weiterhin beständig anwachsen. Auf diesen Mikroprozessoren ist eine immer komplexer werdende Software implementiert, die im Auto Steuerungsfunktionen angefangen von der Zentralverriegelung über die Klimaanlage bis hin zur Motorsteuerung übernimmt. Die so entstehenden Kontrolleinheiten unterstützen den Fahrer in Standard- oder auch sicherheitskritischen Situationen.

Beispiel: Airbag

Ein konkretes Beispiel aus dem Automobilbau ist dabei folgendes: Nahezu alle heute gebauten Kraftfahrzeuge sind mit einem *Airbag-System* ausgestattet. Der Airbag bläst sich im Falle eines Unfalls mit hoher Geschwindigkeit auf und schützt so den Autoinsassen vor dem Aufprall auf das Lenkrad bzw. den Armaturenträger.

Das korrespondierende eingebettete System arbeitet in etwa wie folgt: Im Fahrzeugs sind mehrere Sensoren verbaut, die dessen Beschleunigung messen. Prallt ein Wagen auf ein Hindernis, entsteht eine negative Beschleunigung. Diese wird vom Sensor registriert, der einen Gasgenerator zündet und damit das Aufblasen des Airbags in Gang setzt. Der geschilderte Vorgang passiert in der sehr kurzen Zeit von nur wenigen Hundertstel Sekunden. Die hohe Aufblasgeschwindigkeit des Airbags stellt sicher, dass das Luftkissen bereits voll aufgeblasen ist, wenn der Oberkörper des Autoinsassen nach vorne geschleudert wird. Nach dem selben Prinzip funktionieren auch zusätzliche Airbags für den Seiten-, Brust-, oder Kopfbereich. Hier sind die Sensoren an den jeweiligen Stellen an den Seiten des Wagens angebracht.

Weitere Beispiele

Eingebettete Steuer- und Regelsysteme sind mittlerweile aber nicht nur in Kraftfahrzeugen und Flugzeugen verbaut, sondern in beinahe allen technischen Geräten des täglichen Lebens enthalten (Broy et al., 1998). Im Bereich der Konsumgüterelektronik (Unterhaltungs- und Haushaltsgerätebereich) sind Fotoapparate, Videokameras, HiFi- und TV-Geräte, Set-Top-Boxen (Decoder für digitale TV-Programme), Waschmaschinen, Wäschetrockner, Mikrowellengeräte, Staubsauger sowie Heizungssteuerungen prominente Beispiele für Produkte, die im Allgemeinen mehrere eingebettete Systeme enthalten.

Aus der großen Anzahl der hier aufgelisteten Produktkategorien und deren breiter Streuung über das gesamte Produktspektrum lässt sich bereits die Bedeutung heute im Einsatz befindlicher eingebetteter Systeme erahnen. Hinzu kommt, dass letztere insbesondere in Massenmärkten wie Konsumelektronik, Telekommunikation und Automobiltechnik weite Verbreitung gefunden haben und in immer größerem Umfang eingesetzt werden. Beispielsweise hat sich im Zeitraum zwischen 1985 und 1994 der Programmumfang, also der Software-Anteil, in Automobilsystemen der Firma Siemens Automotive S.A. (Toulouse) alle zwei bis drei Jahre verdoppelt, siehe (Siemens, 1994).

1.4 Begriffsdefinitionen

Zum besseren Verständnis der in diesem Buch verwendeten Begriffe geben wir im Folgenden eine Reihe entsprechender Definitionen an. Dieser Abschnitt ist insbesondere deshalb von Bedeutung, da sowohl im akademischen, als auch im industriellen Umfeld einige der Begriffe mit unterschiedlicher Bedeutung belegt sind. Die hier formulierten Definitionen beziehen sich überwiegend auf die informationsverarbeitenden Systeme (Hardware beziehungsweise Software) und weniger auf die technisch-physikalische Umgebung (Broy et al., 1998):

Definition (System)

Definition (System):
Unter einem System versteht man ein mathematisches Modell S, das einem Eingangssignal der Größe x ein Ausgangssignal y der Größe y=S(x) zuordnet.

Wenn das Ausgangssignal nur vom aktuellen Wert des Eingangssignals abhängt, so spricht man von einem *gedächtnislosen* System. Wenn aber dieser von vorhergehenden Eingangssignalen abhängt, so spricht man von einem *dynamischen* System.

Definition (reaktives System)

Definition (reaktives System):
Ein reaktives System kann aus Software und/oder Hardware bestehen und setzt Eingabeereignisse (deren zeitliches Auftreten meist nicht vorhergesagt werden kann) – oftmals aber nicht notwendigerweise unter Einhaltung von Zeitvorgaben – in Ausgabeereignisse um.

Definition (hybrides System)

Definition (hybrides System):
Systeme, die sowohl kontinuierliche (analoge), als auch diskrete Datenanteile (wertkontinuierlich) verarbeiten und/oder sowohl über kontinuierliche Zeiträume (zeitkontinuierlich), als auch zu diskreten Zeitpunkten mit ihrer Umgebung interagieren, heißen hybride Systeme.

Zur expliziten Unterscheidung zwischen beiden Formen hybrider Systemanteile werden die Begriffe “datenkontinuierlich/-diskret” oder „wertkontinuierlich/-diskret“ und “zeitkontinuierlich/-diskret” verwendet.

Definition (verteiltes System)

Definition (verteiltes System):
Ein verteiltes System besteht aus Komponenten, die räumlich oder logisch verteilt sind und mittels einer Koppelung bzw. Vernetzung zum Erreichen der Funktionalität des Gesamtsystems beitragen.

Die Entwicklung von Steuergeräten hat in der zweiten Hälfte des letzten Jahrhunderts einen bedeutenden Wechsel vollzogen. Angefangen von elektromechanischen Steuergeräten über elektrische Steuergeräte ist man heutzutage bei elektronischen (vollprogrammierbaren) Steuergeräten angelangt.

Definition (Steuergerät)

Definition (Steuergerät):
Ein Steuergerät (engl. Electronic Control Unit, kurz ECU) ist die physikalische Umsetzung eines eingebetteten Systems. Es stellt damit die Kontrolleinheit eines mechatronischen Systems dar. In mechatronischen Systemen bilden Steuergerät und Sensorik/Aktorik oft eine Einheit.

Steuergeräte sind im Prinzip wie folgt aufgebaut: Die Kernkomponente des Steuergeräts stellt ein Mikrocontroller oder Mikroprozessor (Beispiele: Power PC, Alpha PC) dar. Zusätzlich kann es optional ein externes RAM und/oder ROM besitzen sowie sonstige Peripherie und Bauelemente.

Definition (mechatronisches System, Mechatronik)

Definition (mechatronisches System, Mechatronik):
Wird Elektronik zur Steuerung und Regelung mechanischer Vorgänge räumlich eng mit den mechanischen Systembestandteilen verbunden, so sprechen wir von einem mechatronischen System. Der Forschungszweig, der sich mit den Grundlagen und der Entwicklung mechatronischer Systeme befasst, heißt Mechatronik.

Der Begriff *Mechatronik* (engl. mechatronics) ist ein Kunstwort bestehend aus *Mecha*nik und Elek*tronik*. Bei der Mechatronik handelt sich um ein interdisziplinäres Gebiet der Ingenieurwissenschaften, das auf Maschinenbau, Elektrotechnik und der Informatik aufbaut. Mechatronische Systeme enthalten in zunehmendem Maße hierarchisch angeordnete, untereinander gekoppelte eingebettete Systeme. Ein typisches mechatronisches System nimmt Signale auf, verarbeitet sie und gibt wiederum Signale aus, die dann ggf. in Kraft oder Bewegung umgesetzt werden.

1.5 Logischer Aufbau eingebetteter Systeme

In diesem Abschnitt charakterisieren wir die grundlegenden Bestandteile eingebetteter Systeme und erklären die daraus resultierende logische Strukturierung der Anforderungen. Nach der Lektüre dieses Abschnittes sollte der Leser den prinzipiellen logischen Aufbau eines eingebetteten Systems und dessen wesentliche Bestandteile kennen.

Beim Einsatz von Hardware in eingebetteten Systemen ist auf die *Umgebungsbedingungen am Einsatzort* besonders zu achten. Sie muss unter anderem robust gegen folgende Störfaktoren sein: Wärme bzw. Kälte, Staub, Feuchtigkeit, Spritzwasser, mechanische Schwingungen bzw. Stöße, Fremdkörper und elektromagnetische Störungen (EMS). Insbesondere in der Automobilindustrie gibt es hier genaue, oftmals Hersteller-spezifische Vorschriften, die diesbezüglich genaue Vorgaben enthalten. Sie können durch physikalische (z. B. durch Schirmung bzw. Gehäuse) und geometrische (beispielsweise Vermeidung von EMS durch Verdrillen der Kabel) Maßnahmen eingehalten werden.

Die 5 Strukturbestandteile eingebetteter Systeme

Ein eingebettetes System kann in der Regel in die folgenden fünf strukturellen Bestandteile zerlegt werden (Broy und Scholz, 1998):

- Die Kontrolleinheit bzw. das Steuergerät, d. h. das eingebettete Hardware/Software System,
- die Regelstrecke mit Aktoren (auch: Aktuatoren) und Sensoren, d. h. das gesteuerte physikalische System,
- die Benutzerschnittstelle,
- die Umgebung sowie
- den Benutzer.

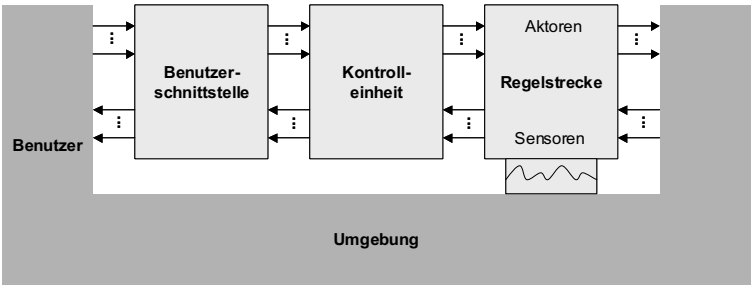

Abb. 1.2: Logische Referenz-architektur eines eingebetteten Systems

Grobarchitektur

Die *Grobarchitektur* eines eingebetteten Systems unterteilt das System in diese Bestandteile und beschreibt, wie sie miteinander verbunden sind. Abbildung 1.2 zeigt diese Architektur als Datenflussdiagramm. Die gerichteten Pfeile stellen gerichtete Kommunikationskanäle dar. Auf diesen Kanälen werden Ströme diskreter Nachrichten oder kontinuierlicher Signale („diskret" bedeutet hier immer ereignisdiskret, „kontinuierlich" meint zeit- und wertkontinuierlich) übermittelt. Die gezackte Linie, welche die Kommunikationskanäle zwischen Regelstrecke und Umgebung markiert, deutet an, dass diese Bestandteile auf sehr komplexe, schwer formalisierbare Weise in Wechselwirkung stehen können. Die Kontrolleinheit ist nicht direkt mit der Umgebung verbunden, sondern nur mit der Benutzerschnittstelle und der Regelstrecke. Die Grauschattierung in der Zeichnung weist darauf hin, dass wir zwischen Benutzer und Systemumgebung nicht klar trennen.

Wirkungskette

Wie wir der Abbildung 1.2 entnehmen können, handelt es sich beim Ablauf eines eingebetteten Systems um eine geschlossene Wirkungskette, vgl. Abbildung 1.3. Die Aufgaben dieser Wirkungskette sind:

- Die Erfassung von Eingabeereignissen durch Sensoren,
- die steuernde Einwirkung auf den zu kontrollierenden Prozess durch Aktoren und
- die Umwandlung (kurz: Wandlung) analoger elektrischer Größen der Eingabeereignisse in digitale Signale zur Verarbeitung durch Rechner (die sogenannte Analog-Digital/Digital-Analog-Wandlung, kurz AD/DA-Wandlung).

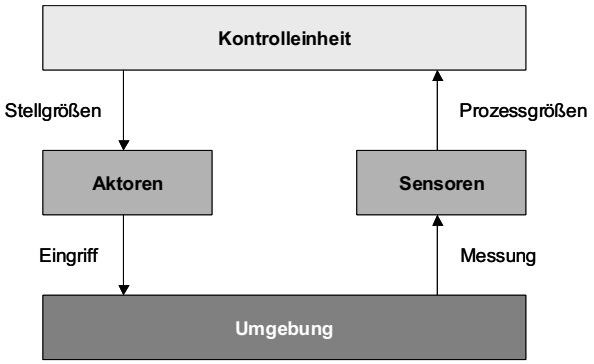

Abb. 1.3: Wirkungskette System/ Umgebung

Der zu steuernde technische Prozess ist also über Sensoren und Aktoren an das Steuergerät gekoppelt und kommuniziert über diese mit ihm. Sensoren und Aktoren fasst man unter dem Begriff *Peripherie* oder *I/O-System* zusammen. I/O (engl. Input/Output) steht hier für Eingabe/Ausgabe (kurz: E/A).

1.5.1 Kontrolleinheit

Die Kontrolleinheit empfängt Signale von der Benutzerschnittstelle und den Sensoren an der Regelstrecke. Diese Eingaben werden verarbeitet und Antwortsignale werden an die Benutzerschnittstelle und die Aktoren an der Regelstrecke geschickt. Die Kontrolleinheit bildet den Kern des eingebetteten Systems. Selbstverständlich muss die Kontrolleinheit keine monolithische Komponente sein. Sie kann im Entwurf durchaus in ein Netzwerk von parallelen, örtlich verteilten Subkomponenten zerlegt werden, die mit Benutzerschnittstelle und Regelstrecke interagieren. Fragen der Parallelität und der Verteilung sind beim Entwurf von eingebetteten Systemen allerdings Teil der sogenannten „Glass-Box-Sicht“ auf die Kontrolleinheit. In den frühen Phasen der Systementwicklung sind wir dagegen hauptsächlich an der „Black-Box-Sicht“ interessiert. Eine direkte Verbindung von Benutzerschnittstelle und Regelstrecke kann als Spezialfall der Kontrolleinheit modelliert werden. Wie wir bereits gesehen haben, stellen Steuergeräte die Kontrolleinheit eines mechatronischen Systems dar.

Moderne PKWs der Oberklasse enthalten inzwischen bis zu 100 Steuergeräten (ECU, Electronic Control Unit), Tendenz steigend. Damit einhergehend werden ca. 3 km Kabel und ca. 2.000 Steckverbindungen verbaut. In den Steuergeräten reguliert Software