

Einleitung

1.1 Begriffe und Abgrenzung

Bis in die späten 80er Jahre war die Informationsverarbeitung mit großen *Mainframe*-Rechnern und riesigen Bandlaufwerken verbunden. Während der 90er Jahre hat sich die Informationsverarbeitung zu den *Personal Computern*, den PCs, verlagert. Dieser Trend zur Miniaturisierung geht weiter, und die Mehrzahl informationsverarbeitender Geräte werden in naher Zukunft kleine, teilweise auch tragbare Computer sein, die in größere Produkte integriert sind. Das Vorhandensein von Prozessoren in diesen umgebenden Produkten, wie z.B. in Telekommunikationsgeräten, wird weniger sichtbar sein als beim klassischen PC. Daher wird dieser Trend als der **verschwindende Computer** bezeichnet. Allerdings bedeutet dieser Begriff nicht, dass die Computer tatsächlich verschwinden werden, sondern vielmehr, dass sie überall sein werden. Diese neue Art von Anwendungen der Informationsverarbeitung wird auch *ubiquitous computing* (**allgegenwärtiges Rechnen**), *pervasive computing* [Hansmann, 2001], [Burkhardt, 2001] oder *ambient intelligence* [Koninklijke Philips Electronics N.V., 2003, Marzano und Aarts, 2003] genannt. Diese drei Begriffe beschäftigen sich mit unterschiedlichen Nuancen der zukünftigen Informationsverarbeitung. *Ubiquitous Computing* konzentriert sich auf die langfristige Zielsetzung, Informationen jederzeit und überall zur Verfügung zu stellen, wohingegen *pervasive computing* sich mehr mit praktischen Aspekten, wie etwa der Ausnutzung bereits vorhandener Technologie, beschäftigt. Im Bereich *ambient intelligence* findet man einen Schwerpunkt auf der Kommunikationstechnologie im Wohnbereich der Zukunft sowie im Bereich der intelligenten Gebäudetechnik. Eingebettete Systeme sind einer der Ausgangspunkte dieser drei Gebiete und sie steuern einen Großteil der notwendigen Technologie bei. **Eingebettete Systeme sind informationsverarbeitende Systeme, die in ein größeres Produkt integriert sind**, und die normalerweise nicht direkt vom Benutzer wahrgenommen werden. Beispiele für eingebettete Systeme sind informationsverarbeitende Systeme in

Telekommunikationsgeräten, in Transportsystemen wie Autos, Zügen, Flugzeugen, in Fabriksteuerungen und in Unterhaltungsgeräten. Diese Systeme haben die folgenden gemeinsamen Charakteristiken:

- Häufig sind eingebettete Systeme mit der physikalischen Umwelt verbunden. **Sensoren** sammeln Informationen über die Umgebung und **Aktuatoren**¹ nehmen Einfluss auf die Umwelt.
- Eingebettete Systeme müssen **verlässlich** sein.

Viele eingebettete Systeme sind sicherheitskritisch und müssen deshalb verlässlich arbeiten. Atomkraftwerke sind ein Beispiel für extrem sicherheitskritische Systeme, die zumindest zum Teil von Software gesteuert werden. Verlässlichkeit ist auch in anderen Systemen wichtig, so etwa in Autos, Zügen, Flugzeugen usw. Ein Hauptgrund, warum diese Systeme sicherheitskritisch sind, ist die Tatsache, dass sie direkt mit ihrer Umgebung in Verbindung stehen und einen unmittelbaren Einfluss auf diese Umgebung haben.

Verlässlichkeit beinhaltet die folgenden Eigenschaften eines Systems:

1. **Zuverlässigkeit:** Die Zuverlässigkeit ist die Wahrscheinlichkeit, dass ein System nicht ausfällt.
 2. **Wartbarkeit:** Die Wartbarkeit ist die Wahrscheinlichkeit, dass ein ausgefallenes System innerhalb einer bestimmten Zeitspanne wieder repariert werden kann.
 3. **Verfügbarkeit:** Die Verfügbarkeit ist die Wahrscheinlichkeit, dass ein System korrekt arbeitet. Sowohl Zuverlässigkeit als auch Wartbarkeit müssen hoch sein, um eine hohe Verfügbarkeit zu erreichen.
 4. **Sicherheit:** Dieser Ausdruck beschreibt die Eigenschaft, dass ein ausfallendes System keinen Schaden verursacht.
 5. **Integrität:** Dieser Begriff beschreibt die Eigenschaft, dass vertrauliche Daten geheim bleiben und dass die Authentizität der Kommunikation gewährleistet ist.
- Eingebettete Systeme müssen **effizient** sein. Die folgenden Größen können dazu verwendet werden, die Effizienz von eingebetteten Systemen zu beschreiben:
 1. **Energie:** Viele eingebettete Systeme sind in tragbare Geräte integriert, die ihre Energie über Batterien beziehen. Nach aktuellen Vorhersagen [SEMATECH, 2003] wird die Kapazität von Batterien nur langsam wachsen. Allerdings wachsen die Anforderungen an die Rechenkapazität, insbesondere für Multimedia-Anwendungen, sehr stark

¹ Aktuatoren sind in diesem Kontext Geräte, die numerische Werte in physikalische Größen umwandeln.

an, und die Kunden erwarten trotzdem lange Batterielaufzeiten. Daher muss die verfügbare elektrische Energie sehr effizient eingesetzt werden.

2. **Codegröße:** Der Code, der auf einem eingebetteten System ausgeführt werden soll, muss innerhalb des Systems selbst gespeichert werden. Typischerweise gibt es in solchen Systemen keine Festplatte, auf der das Programm abgelegt werden kann. Das dynamische Hinzufügen von Code ist (noch) eine Ausnahmeerscheinung und tritt nur z.B. bei Java-Telefonen oder Set-Top-Boxen auf. Aufgrund der Rahmenbedingungen muss die Größe des Codes so klein wie möglich sein, er muss aber dennoch das gewünschte Verhalten zeigen. Dies gilt insbesondere für *Systems On a Chip* (SOCs), bei denen sich alle informationsverarbeitenden Schaltkreise auf einem einzigen Chip befinden. Wenn der Befehlsspeicher auf diesem Chip integriert werden soll, muss er zwangsläufig sehr effizient genutzt werden.
 3. **Laufzeit-Effizienz:** Die gewünschte Funktionalität eines eingebetteten Systems sollte mittels eines minimalen Aufwands an Ressourcen zur Verfügung gestellt werden. Auch Zeitbedingungen sollten unter Verwendung minimaler Hardware- und Energie-Ressourcen sicher eingehalten werden können. Außerdem ist es wichtig, dass wirklich nur notwendige Hardware-Komponenten vorhanden sind. Komponenten, die nicht in der Lage sind, die maximal benötigte Zeit zur Ausführung einer Aufgabe (die sog. *Worst Case Execution Time* (WCET)) zu verbessern, wie z.B. viele Caches können in zeitkritischen Systemen weggelassen werden.
 4. **Gewicht:** Alle tragbaren Geräte müssen ein möglichst geringes Gewicht aufweisen, da dies oft einen Kaufanreiz für ein bestimmtes System darstellt.
 5. **Preis:** Für eingebettete Systeme, die in großen Stückzahlen hergestellt werden, insbesondere solche in Konsumelektronik, ist der Wettbewerb auf dem Markt ein sehr wichtiger Aspekt. Daher müssen sowohl Hardware-Komponenten als auch das Software-Entwicklungs-Budget sorgfältig und effizient genutzt werden.
- Diese Systeme werden meistens **für eine bestimmte Applikation entworfen**. Beispielsweise wird ein Prozessor im Steuergerät eines Autos oder eines Zuges nur immer genau seine Steuerungs-Software ausführen. Niemand würde versuchen, auf einem solchen System ein Computerspiel oder eine Tabellenkalkulation auszuführen. Dafür gibt es zwei Hauptgründe:
 1. Würde man die Ausführung anderer Programme auf diesen Systemen zulassen, würden sie dadurch weniger verlässlich.
 2. Die Ausführung anderer Programme ist nur möglich, wenn entsprechende Ressourcen wie z.B. Speicher ungenutzt sind. In einem effizienten System sollten allerdings alle Ressourcen effizient genutzt werden.

- Die meisten eingebetteten Systeme haben keine Tastatur, keine Maus und keine großen Bildschirme als Benutzerschnittstelle. Stattdessen besitzen sie ein spezialisiertes Benutzer-Interface, das z.B. aus Knöpfen, einem Lenkrad, Pedalen usw. bestehen kann. Aus diesem Grund nimmt der Benutzer das informationsverarbeitende System im Hintergrund kaum wahr. Diese neue Computer-Ära wurde deshalb auch durch das Schlagwort des **verschwindenden Computers** charakterisiert.
- Viele eingebettete Systeme müssen **Echtzeit-Bedingungen** einhalten. Wenn die Berechnungen nicht innerhalb einer festgelegten Zeitspanne durchgeführt werden, kann dies zu schweren Qualitätseinbußen führen (wenn z.B. die Audio- oder Videoqualität durch Aussetzer oder Sprünge leidet), oder es kann sogar zu körperlichem Schaden des Benutzers führen (wenn sich z.B. Systeme in Autos, Zügen oder Flugzeugen nicht wie vorgesehen verhalten). Zeitbedingungen, deren Nichteinhalten zu einer Katastrophe führen kann, heißen **harte Zeitbedingungen** [Kopetz, 1997]. Alle anderen Zeitbedingungen heißen **weiche Zeitbedingungen**.

Viele aktuelle informationsverarbeitende Systeme verwenden Techniken, um die durchschnittliche Geschwindigkeit zu erhöhen. So verbessern etwa Caches die durchschnittliche Leistungsfähigkeit eines Systems. In anderen Fällen wird eine zuverlässige Kommunikation dadurch erreicht, dass bestimmte Nachrichten wiederholt werden. Pakete werden etwa von Internet-Protokollen erneut verschickt, wenn die Originalnachricht verlorengegangen ist. Im Durchschnitt führen solche wiederholten Sendevorgänge (hoffentlich) nur zu einer geringen Verringerung der Übertragungsgeschwindigkeit, obwohl die Verzögerung für eine bestimmte Nachricht Größenordnungen über der normalen Verzögerungszeit liegen kann. Im Bereich der echtzeitfähigen Systeme ist eine Argumentation, die auf durchschnittlicher Leistungsfähigkeit aufbaut, nicht akzeptabel. **Eine garantierte System-Antwortzeit muss ohne statistische Argumente erklärt werden können** [Kopetz, 1997].

- Viele eingebettete Systeme sind **hybride Systeme**. Das bedeutet, sie enthalten sowohl analoge als auch digitale Schaltungsteile. Analoge Teile verarbeiten kontinuierliche Signalwerte in kontinuierlicher Zeit, wohingegen digitale Teile diskrete Signalwerte in diskreten Zeitschritten verarbeiten.
- Typischerweise sind eingebettete Systeme **reaktive Systeme**, die man wie folgt definieren kann: *Ein reaktives System ist ein System, das in kontinuierlicher Interaktion mit seiner Umgebung steht und mit einer Geschwindigkeit arbeitet, die von seiner Umwelt vorgegeben wird* [Bergé et al., 1995]. Reaktive Systeme befinden sich in einem bestimmten Zustand und warten auf eine Eingabe. Nach jeder Eingabe führen sie eine Rechnung durch, erzeugen eine Ausgabe und wechseln in einen neuen Zustand. Aus diesem Grunde lassen sich solche Systeme gut als endliche Automaten modellieren. Reine mathematische Funktionen, die lediglich die von den Algorith-

men gelösten Probleme beschreiben, wären keine geeigneten Modelle zur Beschreibung reaktiver Systeme.

- Eingebettete Systeme sind **in der Lehre und in öffentlichen Diskussionen unterrepräsentiert**. *Um eingebettete Prozessoren wird kein Wirbel im Fernsehen und in Zeitschriftenartikeln gemacht...* [Ryan, 1995]. Ein Problem bei der Vermittlung von Wissen über eingebettete Systeme ist die Ausstattung, die benötigt wird, um das Thema interessant und praktisch erfahrbar zu machen. Außerdem erschwert die hohe Komplexität von praktisch eingesetzten eingebetteten Systemen die Wissensvermittlung.

Wegen dieser gemeinsamen Charakteristiken eingebetteter Systeme erscheint es sinnvoll, gemeinsame Ansätze zum Entwurf solcher Systeme zu betrachten, statt sich nur auf einzelne Anwendungsgebiete und deren Lösung zu konzentrieren.

Natürlich hat nicht jedes eingebettete System alle oben genannten Eigenschaften. Wir können „eingebettete Systeme“ also wie folgt definieren: **Informationsverarbeitende Systeme, welche die meisten der oben aufgezählten Eigenschaften erfüllen, heißen eingebettete Systeme**. Diese Definition beinhaltet eine gewisse Unschärfe. Allerdings erscheint es weder notwendig noch möglich, diese Unschärfe zu vermeiden.

Die meisten der genannten charakteristischen Eigenschaften eingebetteter Systeme finden sich auch in den kürzlich eingeführten Gebieten des *ubiquitous computing* oder *pervasive computing*, auch bekannt als *ambient intelligence*. Diese Begriffe lassen sich auf deutsch mit **allgegenwärtiges Rechnen** übersetzen. Das Hauptziel dieser Gebiete ist es, Informationen **überall und jederzeit** verfügbar zu machen, weswegen sie sich auch mit Kommunikationstechnik beschäftigen. Abb. 1.1 zeigt eine graphische Darstellung, wie *ubiquitous computing* von eingebetteten Systemen und Kommunikationstechnik beeinflusst wird.

Beispielsweise müssen beim *ubiquitous computing* Echtzeit- und Verlässlichkeitsbedingungen von eingebetteten Systemen eingehalten werden, während fundamentale Techniken der Kommunikationstechnik, wie z.B. Netzwerke, verwendet werden.

1.2 Anwendungsgebiete

Die folgende Liste beinhaltet Anwendungsbereiche, in denen eingebettete Systeme eingesetzt werden:

- **Automobilbereich:** Moderne Autos können nur noch verkauft werden, wenn sie einen beträchtliche Anteil an Elektronik enthalten, z.B. Airbag-Steuerungs-Systeme, elektronische Motorsteuerungen, ABS, Klimaanlage, Navigationsgeräte mit GPS-Unterstützung und viele andere.

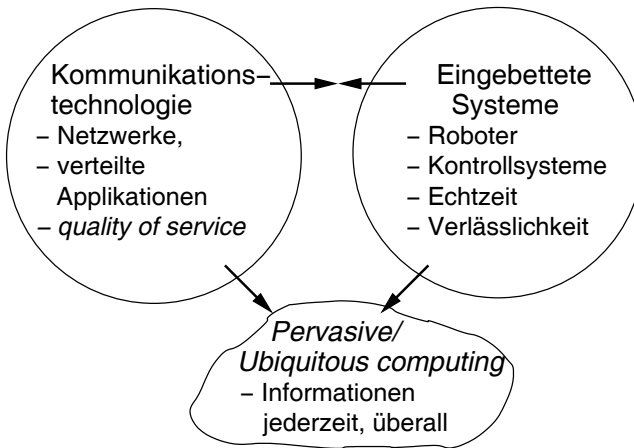


Abb. 1.1. Einfluss eingebetteter Systeme auf *ubiquitous computing*

- **Bordelektronik im Flugzeug:** Einen Großteil des Gesamtwertes eines Flugzeugs machen heute die informationsverarbeitenden Systeme aus. Dazu gehören Flugkontrollsysteme, Anti-Kollisions-Systeme, Piloten-Informationssysteme und andere. Die Verlässlichkeit der Systeme ist in diesem Bereich von allerhöchster Wichtigkeit.
- **Eisenbahntechnik:** Bei Zügen, Lokomotiven und stationären Sicherheitssystemen ist die Situation ähnlich wie bei Autos und Flugzeugen. Auch hier tragen die Sicherheitssysteme einen Großteil zum gesamten Wert bei, und die Verlässlichkeit hat eine ähnlich hohe Priorität.
- **Telekommunikation:** Die Verkaufszahlen von Handys sind in den vergangenen Jahren so stark gestiegen wie in kaum einem anderen Bereich. Schlüsselaspekte bei der Entwicklung von Handys sind das Beherrschen der Sendetechnik, digitale Signalverarbeitung und ein geringer Energieverbrauch.
- **Medizinische Systeme:** Im Bereich der medizinischen Geräte gibt es durch die Verwendung von informationsverarbeitenden Geräten ein großes Innovationspotential.
- **Militärische Anwendungen:** Informationsverarbeitung wird seit vielen Jahren in militärischen Ausrüstungsgegenständen verwendet. Eine der ersten Computeranwendungen war die automatische Auswertung von Radarsignalen.
- **Authentifizierungs-Systeme:** Eingebettete System können zur Benutzer-Authentifizierung verwendet werden. Beispielsweise können neuartige Zahlungssysteme die Sicherheit gegenüber klassischen Systemen deutlich erhöhen. Ein Beispiel für ein solches System ist etwa der SMARTpen® [IMEC, 1997] (s. Abb. 1.2).

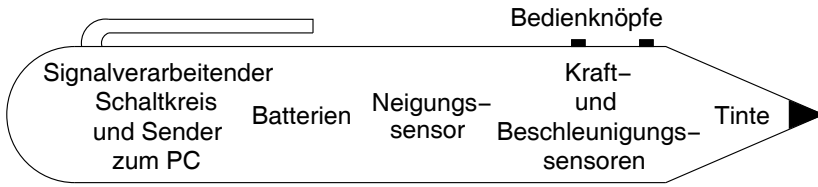


Abb. 1.2. SMARTpen

Der SMARTpen sieht aus wie ein Stift und analysiert während der Unterschrift physikalische Parameter wie z.B. Neigung, Anpressdruck und Beschleunigung. Diese Werte werden an einen Rechner übertragen und mit den Referenzdaten verglichen, die zu dem betreffenden Benutzer gespeichert sind. So kann man sowohl das Aussehen der Unterschrift als auch die Art und Weise, wie sie zu Papier gebracht wurde, mit der gespeicherten Information vergleichen.

Andere Authentifizierungssysteme sind etwa Fingerabdrucksensoren oder Gesichtserkennungs-Systeme.

- **Unterhaltungselektronik:** Video- und Audio-Geräte sind ein besonders wichtiger Sektor der Elektronikindustrie. Die Anzahl informationsverarbeitender Systeme auf diesem Gebiet erhöht sich ständig. Neue Dienste und bessere Qualität werden durch moderne Methoden der digitalen Signalverarbeitung erreicht. Viele Fernseher, Multimedia-Handys und Spielekonsolen beinhalten Hochleistungsprozessoren und -Speichersysteme. Diese stellen eine besondere Gattung eingebetteter Systeme dar.
- **Fabriksteuerungen:** Im Bereich der Fabriksteuerungen werden eingebettete Systeme traditionell seit Jahrzehnten eingesetzt. Die Sicherheit solcher Systeme ist sehr wichtig, wohingegen der Energieverbrauch ein weniger wichtiges Kriterium ist. Das Beispiel in Abb. 1.3 (entnommen aus Kopetz [Kopetz, 1997]) zeigt einen Flüssigkeitsbehälter, der mit einem Abflussrohr verbunden ist. Das Rohr ist mit einem Ventil und einem Sensor versehen. Ein Computer kann die Daten des Sensors verwenden, um die Flüssigkeitsmenge zu steuern, die durch das Rohr abfließt.

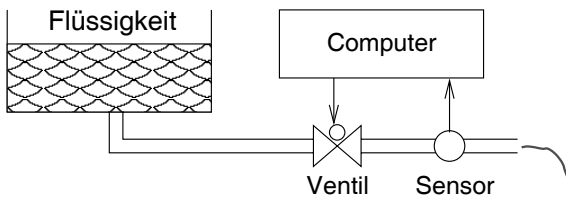


Abb. 1.3. Regelung eines Ventils

Das Ventil ist ein Beispiel für einen Aktuator (Definition s. Seite 2).

- **Intelligente Gebäude:** Informationsverarbeitende Systeme können verwendet werden, um den Komfort in Gebäuden zu verbessern, deren Energieverbrauch zu verringern oder um die Sicherheit zu erhöhen. Vorher unabhängige Teilsysteme müssen zu diesem Zweck miteinander verbunden werden. Der Trend geht dahin, Klimaanlage, Lichtsteuerungen, Zugangskontrollen, Abrechnungssysteme sowie Informationsverteilung und -bereitstellung in ein einziges System zu integrieren. So kann man z.B. Energie sparen, indem man Klimaanlage, Heizung und Beleuchtung herunterfährt, wenn die betreffenden Räume leer sind. Verfügbare Räume können an geeigneten Stellen angezeigt werden, wodurch sowohl die Suche nach einem Raum für ein spontanes Meeting als auch die Aufgabe der Reinigungskräfte vereinfacht wird. Der Geräuschpegel der Klimaanlage kann an die aktuelle Situation im Raum angepasst werden. Eine intelligente Steuerung der Jalousien kann die Beleuchtung und die Nutzung der Klimaanlage optimieren. Für leere Räume können größere Temperaturschwankungen akzeptiert werden, außerdem kann die Beleuchtung entsprechend reduziert werden. Im Notfall kann eine Liste der belegten Räume am Eingang des Gebäudes angezeigt werden (vorausgesetzt, der notwendige Strom steht noch zur Verfügung).

Anfangs werden solche Systeme wohl hauptsächlich in modernen Bürogebäuden zu finden sein.

- **Robotik:** Im Bereich der Robotik werden eingebettete Systeme ebenfalls seit langem eingesetzt. Ein wichtiger Bereich auf diesem Gebiet ist die Mechanik. Viele der oben genannten Charakteristiken treffen auch auf die Robotik zu. Seit kurzem gibt es eine neue Art von Robotern, die Tieren oder sogar dem Menschen nachempfunden sind. Abb. 1.4 zeigt ein Beispiel.

Diese Beispiele zeigen die vielfältigen Formen und Ausprägungen eingebetteter Systeme. Warum ist es sinnvoll, all diese verschiedenen Arten von eingebetteten Systemen in einem einzigen Buch zu beschreiben? Es ist sinnvoll, weil die Informationsverarbeitung in all diesen Systemen sehr ähnlich ist, obwohl die Systeme physikalisch vollkommen unterschiedlich sind.

1.3 Wachsende Relevanz von eingebetteten Systemen

Die Größe des Marktes für eingebettete Systeme kann aus einer Reihe von Perspektiven analysiert werden. Wenn man z.B. die Anzahl der momentan im Betrieb befindlichen komplexen Prozessoren betrachtet, so wurde geschätzt, dass mehr als 90% dieser Prozessoren in eingebetteten Systemen verwendet werden. Viele dieser eingebetteten Prozessoren sind 8-Bit-Prozessoren, trotzdem sind 75% aller 32-Bit-Prozessoren ebenfalls in eingebettete Systeme

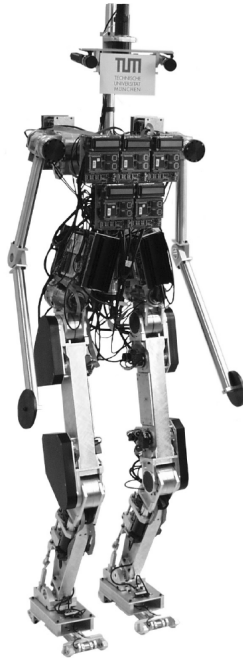


Abb. 1.4. Roboter „Johnnie“ (mit freundlicher Genehmigung von H. Ulbrich, F. Pfeiffer, Lehrstuhl für Angewandte Mechanik, TU München), ©TU München

me integriert [Stiller, 2000]. Bereits 1996 wurde geschätzt, dass der durchschnittliche US-Amerikaner jeden Tag mit 60 Mikroprozessoren in Berührung kommt [Camposano und Wolf, 1996]. Einige Autos der Luxusklasse beinhalten mehr als 100 Prozessoren². Diese Zahlen sind viel größer als man normalerweise annimmt, da vielen Menschen nicht bewusst ist, dass sie Prozessoren verwenden. Wie wichtig eingebettete Systeme sind, ist auch von der Journalistin Mary Ryan geäußert worden [Ryan, 1995]:

„... eingebettete Chips bilden das Rückgrat der von Elektronik getriebenen Welt, in der wir leben. ... sie sind in fast allem enthalten, was mit Elektrizität betrieben wird.“

Vielen Vorhersagen zufolge wird der Markt für eingebettete Systeme bald viel größer sein als der Markt für PC-ähnliche Systeme. Außerdem wird die Menge an Software, die in eingebetteten Systemen verwendet wird, stark zunehmen. Nach Vaandrager wird sich *„für viele Produkte im Bereich der Unterhaltungselektronik die Größe des Codes alle zwei Jahre verdoppeln“* [Vaandrager, 1998].

² Quelle: persönliche Kommunikation

Eingebettete Systeme bilden die Basis der sogenannten *post PC era*, in der sich die Informationsverarbeitung immer weiter weg von PCs, hin zu eingebetteten Systemen verlagert.

Die wachsende Zahl von Anwendungen macht es notwendig, Methoden zu entwickeln, die den Entwurf eingebetteter Systeme unterstützen. Die momentan verwendeten Technologien und Programme weisen noch starke Einschränkungen auf. So besteht ein großer Bedarf an besseren Spezifikations-sprachen, an Programmen, die aus einer Spezifikation eine Implementierung ableiten, an Programmen, welche die Einhaltung von Zeitbedingungen prüfen können, an Echtzeitbetriebssystemen, an Entwurfsmethoden für energiesparende Systeme sowie an Entwurfsmethoden für verlässliche Systeme. Dieses Buch soll Grundlagen zu den wichtigsten Themen vermitteln und ein Ausgangspunkt für weitere Forschungsarbeiten auf diesem Gebiet sein.

1.4 Struktur dieses Buches

Viele der vorhandenen Bücher über eingebettete Systeme erläutern und beschreiben den Einsatz von Mikrocontrollern, sowie die in Mikrocontrollern verwendeten Speicher, die Ein- und Ausgabeeinheiten sowie die Struktur der Interrupts. Es gibt zahlreiche solcher Bücher, etwa [Ganssle, 1992], [Ball, 1996], [Ball, 1998], [Barr, 1999] und [Ganssle, 2000].

Aufgrund der steigenden Komplexität eingebetteter Systeme muss dieser Blickwinkel erweitert werden und zumindest verschiedene Spezifikations-sprachen, Hardware-/Software-Codesign, Compiler-Techniken, Scheduling (Ablaufplanung) sowie Validierungstechniken enthalten. Diese Gebiete werden im vorliegenden Buch behandelt. Das Ziel dieses Buches ist es, Studierenden eine Einführung in das Gebiet der eingebetteten Systeme zu geben und ihnen damit die Möglichkeit zu eröffnen, die verschiedenen Teilgebiete einzuordnen.

Als weiterführende Literatur empfehlen wir folgende Quellen (die teilweise auch während der Vorbereitung dieses Buches herangezogen wurden):

- Es gibt eine Vielzahl von Informationsquellen zu Spezifikations-sprachen. Zu nennen sind frühe Bücher von Young [Young, 1982], Burns und Wellings [Burns und Wellings, 1990] sowie von Bergé [Bergé et al., 1995]. Über neuere Sprachen wie beispielsweise Java, SystemC [Müller et al., 2003], SpecC [Gajski et al., 2000] usw. gibt es sehr viele Bücher.
- Einige Ansätze zum Entwurf und zum Einsatz von Echtzeitbetriebssystemen (*Real-Time Operating Systems* (RTOS)) werden im Buch von Kopetz [Kopetz, 1997] vorgestellt.
- Echtzeit-Scheduling (Ablaufplanung) wird umfassend in den Büchern von Buttazzo [Buttazzo, 2002] und Krishna und Shin [Krishna und Shin, 1997] abgehandelt.

- Die Vorlesungsskripte von Rajiv Gupta [Gupta, 1998] geben einen Überblick über eingebettete Systeme.
- Das Gebiet der Robotik ist eng mit den eingebetteten Systemen verwandt. Wir empfehlen für Informationen auf diesem Gebiet die Bücher von Fu, Gonzalez und Lee [Fu et al., 1987].
- Weitere Informationen findet man in einem Buch von Vahid [Vahid, 2002], in der ARTIST Roadmap [Bouyssounouse und Sifakis, 2005] sowie im forschungsorientierten *Embedded Systems Handbook* [Zurawski, 2006].

Die Struktur dieses Buches entspricht dem vereinfachten Informationsfluss beim Entwurf eingebetteter Systeme, wie er in Abb. 1.5 abgebildet ist.

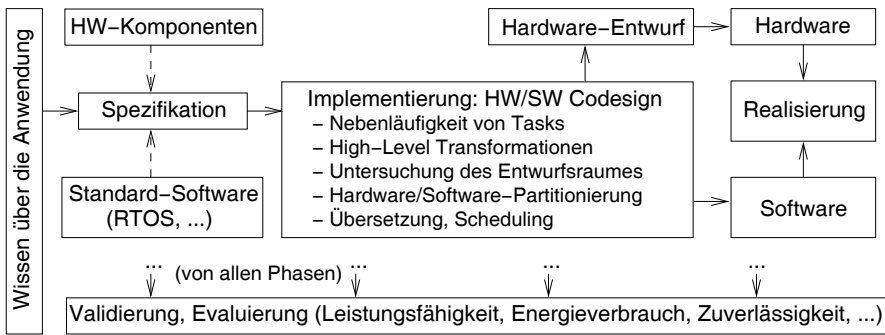


Abb. 1.5. Vereinfachter Informationsfluss beim Entwurf eingebetteter Systeme

Der Informationsfluss beginnt mit einer Idee. Diese Idee muss in einer Entwurfsspezifikation beschrieben werden. Hierbei sollten zur Verfügung stehende Hard- und Software-Komponenten wiederverwendet werden.

Der eigentliche Entwurf beginnt bei der Spezifikation. Typischerweise müssen beim Entwurf eingebetteter Systeme sowohl Hardware- als auch Software-Komponenten in Betracht gezogen werden. Die Entwurfsaktivitäten beinhalten das Abbilden von durchzuführenden Operationen auf nebenläufige Tasks, *High-Level-Transformationen* (etwa komplexe Schleifentransformationen), das Abbilden von Aufgaben auf entweder Hardware oder Software (die sogenannte Hardware/Software-Partitionierung), Untersuchen und Evaluieren des Entwurfsraumes, Software-Übersetzung mit Hilfe von Compilern, sowie Scheduling (Ablaufplanung). Eventuell ist es notwendig, spezialisierte Hardware oder optimierte Prozessorarchitekturen zu verwenden. Der Entwurf dedizierter Hardware wird in diesem Buch allerdings nicht behandelt. Zur Übersetzung der Software in die Maschinsprache können Standard-Compiler verwendet werden. Allerdings bieten diese häufig keine spezielle Unterstützung für eingebettete Systeme. Daher werden wir auch kurz auf Compiler-Techniken eingehen, mit denen die gewünschte Effizienz der generierten Software er-

reicht werden kann. Sobald der Maschinen-Code für jede Teilaufgabe generiert wurde, kann das Scheduling präzise durchgeführt werden. Schließlich werden Software- und Hardware-Beschreibungen zu einer kompletten Beschreibung des Systems zusammengefügt. Anhand dieser Beschreibung kann dann die Produktion des Systems erfolgen.

Nach momentanem Stand der Technik kann man für keinen dieser Schritte garantieren, dass er immer korrekte Ergebnisse liefert. Daher ist es notwendig, den Entwurf zu validieren. Die Validierung besteht darin, die Beschreibung von Zwischenschritten oder des endgültigen Designs mit den anderen Beschreibungen zu vergleichen und auf Abweichungen zu überprüfen. Außerdem muss zu verschiedenen Zeitpunkten der Entwurf evaluiert werden, z.B. in Bezug auf seine Leistungsfähigkeit, seine Verlässlichkeit, Energieverbrauch, Herstellbarkeit usw.

Zu beachten ist, dass Abb. 1.5 den Fluss von **Informationen über das Entwurfsobjekt** darstellt. Die Abfolge von **Entwurfsaktivitäten** muss zu diesem Fluss konsistent sein. Das bedeutet allerdings nicht, dass die Entwurfsaktivitäten einem einfachen Pfad von der Idee hin zum fertigen Produkt entsprechen. In der Praxis müssen einige Aktivitäten wiederholt werden. So kann es notwendig sein, zu einem späteren Zeitpunkt auf die Ebene der Spezifikation zurückzugehen, um zusätzliche Informationen über die Anwendung zu erhalten. Es kann auch vorkommen, dass andere Echtzeit-Betriebssysteme in Betracht gezogen werden müssen, da das ursprünglich vorgesehene Betriebssystem die Anforderungen nicht erfüllt.

Die Kapitel dieses Buches sind analog zum Informationsfluss strukturiert: in Kapitel 2 werden Spezifikationssprachen behandelt. Wichtige Hardware-Komponenten eingebetteter Systeme werden in Kapitel 3 betrachtet. Kapitel 4 widmet sich der Beschreibung von Echtzeit-Betriebssystemen und sogenannter *Middleware* sowie Scheduling-Techniken. Standard-Entwurfstechniken zur Implementierung eingebetteter Systeme, u.a. Compiler-Techniken, werden in Kapitel 5 behandelt. Das letzte Kapitel schließlich befasst sich mit der Evaluation und Validierung.