Kapitel 1

Einleitung

In diesem Kapitel werden wir einige grundlegende Begriffe, die im Zusammenhang mit eingebetteten Systemen benutzt werden, zusammen mit der geschichtlichen Entwicklung vorstellen, sowie auf Möglichkeiten, Herausforderungen und gemeinsame Eigenschaften von eingebetteten und cyber-physikalischen Systemen eingehen. Außerdem werden Aspekte der Ausbildung, Entwurfsabläufe und die Struktur dieses Buches vorgestellt werden.

1.1 Geschichte der Begriffe

Unter dem Begriff „Informationsverarbeitung“ stellte man sich bis zum Ende der achtziger Jahre des letzten Jahrhunderts unweigerlich Großrechner und riesige Bandlaufwerke vor. Die Miniaturisierung ermöglichte später die Einführung der Informationsverarbeitung mit Hilfe von PCs (engl. *Personal Computers*). Büroanwendungen dominierten, aber einige Rechner wurden auch für Steuerungen und Regelungen innerhalb von Rückkopplungsschleifen in der physischen Umgebung eingesetzt.

Später schuf Mark Weiser den Begriff des **allgegenwärtigen Rechnens** (engl. *ubiquitous computing*) [572]. Dieser Begriff bezieht sich auf Weisers Vorhersage, künftig Rechner und Information überall und zu jeder Zeit zur Verfügung zu haben. Weiser sagte auch vorher, dass Rechner künftig in Produkte integriert werden würden, sodass sie unsichtbar werden würden. Daher schuf er den Begriff des **unsichtbaren Computers**. In ähnlicher Weise sorgte die vorhergesagte Durchdringung unseres täglichen Lebens mit rechnenden Geräten zur Einführung der Begriffe **durchdringendes Rechnen** (engl. *pervasive computing*) und **umgebende Intelligenz** (engl. *ambient intelligence*). Die drei Begriffe legen das Schwergewicht auf nur leicht unterschiedliche Aspekte künftiger Informationstechnologie. Der Begriff „allgegenwärtiges Rechnen“ ist verbunden mit dem langfristigen Ziel, Information jederzeit und überall bereitzustellen, wohingegen der Begriff „durchdringendes Rechnen“ mehr mit den praktischen Aspekten und der Nutzung der bereits vorhandenen Technologie assoziiert wird. Der Begriff „umgebende Intelligenz“ wird

in der Regel mit der Nutzung der Informationstechnologie in künftigen Wohnhäusern und intelligenten Bürogebäuden in Verbindung gebracht. Ein Teil der Visionen ist durch die Verbreitung von kleinen mobilen Geräten in Kombination mit dem mobilen Internet bereits Wirklichkeit geworden und diese Verbreitung ist durchaus „durchdringend“ in dem Sinne, dass sie viele Bereiche des täglichen Lebens bereits beeinflusst hat. Von der „Künstlichen Intelligenz“ wird ebenfalls ein großer Einfluss erwartet.

Dabei wird häufig nicht bewusst wahrgenommen, dass die Miniaturisierung auch die Integration der Informationsverarbeitung mit der Umgebung vorangebracht hat. Ein derart integriertes informationsverarbeitendes System kann als eingebettetes System (ES) gemäß nachfolgender Definition bezeichnet werden.

Definition 1.1 (Marwedel [372]): „Eingebettete Systeme sind informationsverarbeitende Systeme, die in umgebende Produkte integriert sind.“

Beispielsweise finden wir eingebettete Systeme in Autos, Schienenfahrzeugen, Flugzeugen, der Telekommunikation und bei der Fertigungsautomatisierung. Angekündigte Produkte wie selbstfahrende Autos und Schienenfahrzeuge machen dabei klar, dass die Miniaturisierung bei eingebetteten Systemen ähnliche Veränderungen erwarten lässt wie die Verfügbarkeit von Mobilfunkgeräten. Für die genannten Beispiele gibt es dabei eine Vielzahl ähnlicher Anforderungen, wie u.a. ein vorhersagbares Echtzeitverhalten, Verlässlichkeit und Effizienz. Für diese Systeme ist der Bezug zur Umgebung sehr wichtig. Dieser Bezug wird in dem nachfolgenden Zitat betont [330]:

“Embedded software is software integrated with physical processes. The technical problem is managing time and concurrency in computational systems”.

Obiges Zitat könnte als Definition des Begriff „eingebettete Software“ benutzt werden und könnte zu einer Definition des Begriffs „eingebettete Systeme“ verallgemeinert werden, indem wir einfach das Wort „Software“ durch das Wort „System“ ersetzen.

Der starke Bezug zur realen Umgebung wurde noch mehr betont durch die Einführung des Begriffs „cyber-physikalisches System“. Dieser kann wie folgt definiert werden.

Definition 1.2 (Lee [331]): „Cyber-physikalische Systeme (engl. *Cyber-Physical Systems* (CPS)) bestehen aus der Integration von Berechnungen und physikalischen Prozessen.“

Eine Betonung der angesprochenen Integration ist sinnvoll, denn in einer Welt voller Anwendungen auf Servern, PCs und Mobilfunkgeräten wird sie häufig ignoriert. Für cyber-physikalische Systeme können wir erwarten, dass Modelle auch die Umgebung beschreiben. In diesem Sinne können wir uns vorstellen, dass cyber-physikalische Systeme eingebettete Systeme (d.h. den informationsverarbeitenden Teil) wie auch die (dynamische) physische Umgebung enthalten, oder CPS = ES + (dynamisches) physisches bzw. physikalisches System. Dies spiegelt sich auch in Abb. 1.1 wider.

Abb. 1.1 Beziehung zwischen eingebetteten Systemen und physischer Umgebung

In ihrem Aufruf für Projektvorschläge nennt die *National Science Foundation* (NSF) in den USA dabei auch die Verbindung (und damit die Kommunikation) verteilter Systeme (gemäß [413] in deutscher Übersetzung): „Die sich herausbildenden cyber-physikalischen Systeme werden koordiniert, verteilt und verbunden sein und sie müssen robust und reagierend sein.“

Vernetzung (und damit die Kommunikation) wird auch im acatech-Bericht über CPS deutlich angesprochen (gemäß [6] in deutscher Übersetzung): Cyber-physikalische Systeme „stellen vernetzte, Software-intensive eingebettete Systeme in einer Regelschleife dar, die vernetzte und verteilte Dienste anbieten.“

In einem Aufruf für Projektvorschläge der Europäischen Kommission [155] werden Verbindungen und Zusammenarbeit von Komponenten erwähnt: „Cyber-physikalische Systeme (CPS) beziehen sich auf die nächste Generation von IKT-Systemen, die verbunden und zusammenarbeitend sind – auch über das Internet der Dinge – und die Bürgern und Firmen einen breiten Bereich von neuartigen Anwendungen und Diensten anbieten.“

Dabei hat die Europäische Kommission bereits früher die Bedeutung der Kommunikation hervorgehoben, wie es in Abb. 1.2 zu sehen ist.

Abb. 1.2 Bedeutung der Kommunikation
(© Europäische Kommission)

Aus diesen Zitaten wird klar, dass die Autoren unter dem Begriff CPS nicht nur die Integration der Cyber- und der physischen Welt verstehen, sondern dass es auch einen starken Kommunikationsaspekt gibt. Tatsächlich wird der Begriff CPS nicht immer konsistent benutzt: einige Autoren betonen die Integration mit der physischen Umgebung, andere betonen die Kommunikation.

Die Kommunikation wird noch expliziter in dem Begriff des „Internets der Dinge“ (engl. *Internet of Things* (IoT)), der wie folgt definiert werden kann:

Definition 1.3 ([185]): Das **Internet der Dinge** „beschreibt die um sich greifende Gegenwart einer Vielfalt von Geräten – wie Sensoren, Aktuatoren und Mobiltelefonen –, die mittels eindeutiger Adressierung in der Lage sind, miteinander zu interagieren und zu kooperieren, um gemeinsame Ziele zu erreichen.“

Der Begriff verbindet die physische Umgebung mit dem Internet, um Sensoren-Information im Internet verfügbar zu machen und um Aktuatoren aus dem Internet heraus zu steuern. Es wird erwartet, dass das Internet der Dinge die Kommunikation zwischen Milliarden von Geräten erlauben wird. Diese Vision beeinflusst viele Industriezweige.

Die Nutzung der IoT-Technologie für die Produktionstechnik ist als „Industrie 4.0“ [68] bezeichnet worden. Ziel ist eine flexiblere Produktion, bei welcher der gesamte Lebenszyklus durch diese Technologie unterstützt wird.

Insgesamt bleibt es etwas dem Einzelnen überlassen, die Verbindung zwischen physischen Umgebungen und der Cyber-Welt als CPS oder als IoT zu bezeichnen. Wir können aber erwarten, dass CPS und IoT gemeinsam betrachtet den größten Teil der künftigen Anwendungen der Informationstechnologie ausmachen, insbesondere wenn wir die Systeme mit intelligenten Algorithmen ausstatten.

Der Entwurf dieser künftigen Anwendungen setzt die Kenntnis der fundamentalen Entwurfstechniken für eingebettete Systeme voraus. Dieses Buch behandelt genau diese fundamentalen Techniken, indem es allgemeine Schnittstellen zwischen den eingebetteten Systemen und der CPS-oder IoT-Umgebung vorstellt. Diese fundamentalen Techniken werden beim Entwurf von CPS- und IoT-Systemen benötigt, auch wenn das nicht immer wieder in den einzelnen Kapiteln wiederholt wird. Für anwendungsspezifische Information sei jedoch auf spezialisiertere Quellen verwiesen.

1.2 Potential

Die nachfolgende Liste verdeutlicht das riesige Potential für Anwendungen der Informationsverarbeitung in CPS- und IoT-Systemen und sie gibt einen Eindruck von der Variationsbreite der verschiedenen Bereiche:

- **Transportwesen und Mobilität:**

- **Automobilelektronik:** Moderne Autos sind in technisch fortgeschrittenen Ländern nur verkäuflich, wenn sie über eine umfangreiche Ausstattung an elektronischen Komponenten verfügen [416]. Beispiele dafür sind *Airbag*-Steuergeräte, Motorsteuerungen, Navigationssysteme, Antiblockiersysteme (ABS), elektronische Stabilitätsprogramme (ESP), Klimaanlage, Diebstahlschutz, Fahrer-Assistenzsysteme und viele mehr. Ein Trend geht hin zu stärker autonom fahrenden Autos. Eingebettete Systeme können nicht nur den Komfort und die Sicherheit eines Autos verbessern, sie können potentiell auch dazu beitragen, die Umweltbelastung von Automobilen zu reduzieren. E-Mobilität ist ohne umfangreiche elektronische Ausstattung nicht vorstellbar.

- **Luftfahrt:** Ein nennenswerter Anteil der Gesamtkosten von Flugzeugen wird für informationsverarbeitende Systeme, wie z.B. Flugkontrollsysteme, Kollisionsvermeidungssysteme, Piloteninformationssysteme, Autopiloten usw. aufgewendet. Die Verlässlichkeit ist von äußerster Wichtigkeit¹. Eingebettete Systeme können die von Flugzeugen verursachten Emissionen (wie z.B. Kohlendioxid) reduzieren. Autonomes Fliegen wird auch mehr und mehr zu einer Realität, zumindest für spezielle Anwendungsbereiche.
- **Schienenverkehr:** Die Lage im Eisenbahnbereich ist ähnlich der Situation bei Autos und Flugzeugen. Auch hier machen Sicherheitseinrichtungen einen beträchtlichen Anteil der Gesamtkosten aus und die Verlässlichkeit ist ausgesprochen wichtig. Neue Signaltechniken zielen darauf, den Zugverkehr bei hohen Geschwindigkeiten und kurzen Abständen zwischen den Zügen zu sichern. Das *European Train Control System* (ETCS) [444] ist ein Schritt in diese Richtung. Autonomer Schienenverkehr wird in begrenzten Kontexten (wie z.B. bei *Shuttle*-Zügen auf Flughäfen) bereits eingesetzt.
- **Maritime Systeme und Schiffbau:** CPS- und IoT-Systeme finden auch im Bereich moderner Schiffe und anderer maritimer Systeme Anwendung, z.B. für die Navigation, für Sicherheitskonzepte, für allgemeine Betriebsoptimierung und für die Dokumentation der Abläufe (siehe z.B. <http://www.smtcsingapore.com/> und <https://dupress.deloitte.com/dup-us-en/focus/internet-of-things/iot-in-shipping-industry.html>).
- **Neue Mobilitätskonzepte:** Der Einsatz von Informationstechnologie und deren Komponenten ermöglicht neue Mobilitätskonzepte. *E-Bikes* lassen auch ungeübte Personen größere Strecken per Fahrrad überwinden. Das subtile Wechselspiel zwischen größerer menschlicher Muskelkraft und Kraft des Elektromotors macht *E-Bikes* zu einem Musterbeispiel für cyber-physikalische Systeme, ähnlich wie E-Roller. Das abendliche Einsammeln von E-Rollern anhand einer Standortliste im Internet gibt dem Begriff „Internet der Dinge“ eine besonders anschauliche Bedeutung. Auch bei neuen Sammelbus-, Mietwagen- und Fahrdienst-Konzepten sind CPS- und IoT-Techniken für eine Realisierung wichtig.
- **Maschinenbau** (einschl. Fabrikautomatisation): Maschinen und Fertigungsanlagen sind Bereiche, in denen eingebettete/cyber-physikalische Systeme traditionell schon seit Jahrzehnten zum Einsatz kommen. CPS- und IoT-Technologie kann genutzt werden, um die Produktion weiter zu optimieren und sie ist ein Schlüssel zu einer flexibleren Produktion, die das Ziel der „Industrie 4.0“-Initiativen ist [68]. Die weitere Automatisierung wird durch die Logistik ermöglicht, die ihrerseits hierzu CPS- und IoT-Technologien einsetzt [298]. Es gibt viele Möglichkeiten, CPS- und IoT-Technologien in der Logistik einzusetzen. Beispielsweise erlaubt die *Radio Frequency Identification* (RFID)-Technologie die einfache Identifikation jedes Objektes, über Einbindung in ein Netz ggf. weltweit. Die mobile Kommunikation erlaubt eine nie gekannte Interaktion aller an einer Fertigung Beteiligten.

¹ Das Beispiel der Boeing 737 MAX [420] macht die Gefahren sehr deutlich.

- **Robotik:** Die Robotik ist ein weiterer traditioneller Einsatzbereich für eingebettete/cyber-physikalische Systeme. Bei Robotern sind insbesondere die mechanischen Aspekte von Interesse. Es wurden Roboter nach dem Vorbild von Menschen oder Tieren entwickelt. Abb. 1.3 zeigt einen solchen Roboter.

Abb. 1.3 Humanoider Roboter „Lola“
(mit Genehmigung durch D. Rixen,
Lehrstuhl für Angewandte Mechanik,
TU München), © TU München

- **Energieversorgung und das intelligente Netz:** In der Zukunft wird die elektrische Energieversorgung voraussichtlich sehr viel stärker dezentralisiert sein. Dies macht es schwer, die Stabilität der Stromversorgung sicherzustellen. Nur mit IKT-Systemen kann es gelingen, eine ausreichende Stabilität zu erreichen. Informationen zur intelligenten Versorgung mit elektrischer Energie können beispielsweise unter den Adressen https://www.smartgrid.gov/the_smart_grid und <http://www.smartgrids.eu/> gefunden werden.
- **Bauingenieurwesen:** CPS-Geräte können im Bauingenieurwesen eingesetzt werden. Beispielsweise kann die Statik von natürlichen und künstlichen Strukturen (z.B. von Bergen, Vulkanen, Brücken und Dämmen) überwacht werden. Die Abb. 1.4 zeigt beispielsweise den Damm der Möhnetalsperre. Mit Hilfe von CPS-Technologie ist es möglich, bei Gefahren rechtzeitige Warnungen zu verbreiten².
- **Intelligente Gebäude:** Intelligente Gebäude sind einer der Bereiche des Bauingenieurwesens. Informationsverarbeitung kann verwendet werden, um den Komfort in Gebäuden zu verbessern, den Energieverbrauch von Gebäuden zu senken und die Sicherheit zu erhöhen. Dafür müssen bisher nicht zusammenhängende Teilsysteme miteinander verbunden werden. Der Trend geht dahin, die Klimasteuerung, die Beleuchtung, die Zugangskontrolle sowie die Verwaltung und Verteilung von Informationen in ein einziges System zu integrieren. Damit können

² Der Dammbruch von Brumadinho (siehe https://de.wikipedia.org/wiki/Dammbruch_von_Brumadinho) zeigt, wie es nicht ablaufen soll.

Abb. 1.4 Beispiel eines zu beobachtenden Dammes, des Mühnesee-Dammes ©P. Marwedel

z.B. Klimaanlage in leeren Räumen mit geringerer Leistung betrieben werden und gleichzeitig die Beleuchtung nur bei Bedarf aktiviert werden. Die durch die Klimatechnik erzeugten Geräusche können automatisch auf das notwendige Minimum reduziert werden. Die intelligente Verwendung von Verdunkelungen kann zudem Beleuchtungs- und Klimaverhältnisse optimieren. Freie Räume können an zentralen Stellen im Gebäude angezeigt werden, damit können die Planung von kurzfristig anberaumten Besprechungen oder die Reinigung vereinfacht werden. In Notfallsituationen kann eine Liste von belegten Räumen am Gebäudeeingang angezeigt werden (solange der Strom nicht ausgefallen ist). Durch diese Maßnahmen lässt sich Energie für Kühlung, Heizung und Beleuchtung einsparen und zudem kann die Sicherheit des Gebäudes verbessert werden. Anfangs werden solche Systeme nur in hochtechnisierten Bürogebäuden vorhanden sein. Ein Trend hin zu energieeffizienten Gebäuden ist aber auch beim Entwurf von Wohnhäusern zu erkennen. Eines der Ziele ist dabei der Entwurf sogenannter **Niedrigenergiehäuser** bzw. **Nullenergiehäuser** (Gebäude, die so viel Energie erzeugen, wie sie verbrauchen) [427]. Solche Gebäude könnten einen deutlichen Beitrag zur Verringerung der globalen Kohlendioxid-Emissionen und damit der globalen Erwärmung leisten.

- **Agrartechnik:** Es gibt viele IoT-Anwendungen in der Landwirtschaft, wie beispielsweise die Sicherstellung der Nachvollziehbarkeit des Lebensweges von Nutztieren. Im Fall des Ausbruchs ansteckender Krankheiten können so unmittelbar Maßnahmen getroffen werden [516]³. Auch kann der Einsatz von Düngemitteln mittels IoT-Technologie dem Boden angepasst werden.
- **Gesundheitswesen:** Produkte des Gesundheitswesens gewinnen insbesondere durch die zunehmende Alterung der Bevölkerung an Bedeutung. Das Potential zur Innovation beginnt bei neuen Sensoren, die Krankheiten schneller und verlässlicher detektieren können. Neue **Datenanalysetechniken** können benutzt werden, um erhöhte Risiken zu erkennen und die Chancen auf Heilung zu verbessern. Dafür kommen i.d.R. auch neue Algorithmen zum Einsatz, beispielsweise solche der

³ Die Bedeutung der Nachvollziehbarkeit von Kontakten im Allgemeinen, über die Tierzucht hinaus, wurde in der Corona-19-Krise besonders deutlich.

Künstlichen Intelligenz (KI). KI bietet auch die Möglichkeit einer personalisierten Medizin, um die Therapien durch angepasste Medikamente zu unterstützen. Neue Geräte können beispielsweise Behinderten helfen. Auch können Operationen roboterassistiert durchgeführt werden. CPS- und IoT-Technologien können dazu dienen, Ergebnisse einer Therapie besser zu überwachen und damit Ärzten die Möglichkeit zu geben, die Wirksamkeit einer Behandlung zu überprüfen. Diese Überwachung schließt auch entfernt lebende Patienten ein. Verfügbare Daten können in Patienteninformationssystemen gespeichert werden. Listen von Projekten in diesem Bereich können beispielsweise unter <http://cps-vo.org/group/medical-cps> und unter <http://www.nano-tera.ch/program/health.html> gefunden werden.

- **Wissenschaftliche Experimente:** Viele der gegenwärtigen wissenschaftlichen Experimente, insbesondere in der Physik, erfordern eine Erfassung der experimentellen Daten mit IKT-Geräten, d.h. der Cyber-Welt. Die Kombination von physikalischen Experimenten und der Cyber-Welt kann als Spezialfall cyberphysikalischer Systeme gesehen werden.
- **Öffentliche Sicherheit:** Das Interesse an Maßnahmen zur Steigerung der öffentlichen Sicherheit wächst im Angesicht von Gewalttaten und Pandemien. CPS- und IoT-Systeme können die öffentliche Sicherheit stärken. Dies schließt die Erkennung oder Authentisierung von Personen über verschiedene Sensoren ein.
- **Katastrophenschutz:** Im Fall größerer Katastrophen wie Erdbeben oder Überflutungen ist es wesentlich, Leben zu retten und Überlebenden zu helfen. Hierzu sind flexible Kommunikationsstrukturen erforderlich.
- **Militärische Anwendungen:** Informationsverarbeitende Komponenten sind seit vielen Jahren ein wichtiger Bestandteil militärischer Systeme. Einige der allerersten Rechner wurden verwendet, um militärische Radarsignale zu analysieren. Sie sind auch heute wichtiger Teil von Abwehrsystemen.
- **Telekommunikation:** Die Mobiltelefonie ist einer der am stärksten wachsenden Märkte der vergangenen Jahre. Wichtige Aspekte von Mobiltelefonen sind der Entwurf der Funkkomponenten, die digitale Signalverarbeitung und ein niedriger Energieverbrauch. Telekommunikation ist auch ein wesentlicher Aspekt für IoT-Technologien.
- **Unterhaltungselektronik:** Video- und Audiosysteme sind ein wichtiger Bereich der Elektronikindustrie. Diese Geräte verwenden einen stetig ansteigenden Anteil an informationsverarbeitenden Komponenten. Neue Dienste und eine gesteigerte Qualität werden dabei durch neuartige Methoden der digitalen Signalverarbeitung ermöglicht. Viele (vor allem hochauflösende) Fernseher, *Smartphones* und Spielkonsolen verwenden leistungsstarke Prozessoren und Speichersysteme. Diese stellen eine spezielle Klasse eingebetteter Systeme dar. Die Betriebssicherheit und Realzeitbedingungen sind bei dieser Klasse nicht ganz so dringend zu berücksichtigen wie bei anderen Klassen. Allerdings sind teilweise auch Realzeitbedingungen einzuhalten, beispielsweise um eine bestimmte Bildwiederholrate zu erreichen oder um Zeitbedingungen in Kommunikationsprotokollen zu garantieren. Auch sind Ressourcen wie die elektrische Energie oder die Kommunikationsbandbreite gerade bei mobilen Geräten extrem wichtig und damit liegt

hinsichtlich der Verfügbarkeit von Ressourcen eine Situation ähnlich der anderer eingebetteter Systeme vor.

Die große Menge von Beispielen zeigt eindrucksvoll die große Vielfalt von eingebetteten Systemen, die in CPS oder IoT-Systemen eingesetzt werden. Weitere Anwendungen werden in einem Bericht zu den Möglichkeiten und den Herausforderungen des Internets der Dinge aufgeführt [516]. Wir gehen davon aus, dass viele der künftigen Anwendungen der IKT-Technologie sich in solchen Systemen finden werden. Aus obiger Liste schließen wir, **dass praktisch alle Ingenieursdisziplinen betroffen sein werden.**

Die lange Liste der Anwendungen führt auch zu einer entsprechenden ökonomischen Bedeutung dieser Systeme. Laut acatech-Bericht [6] wurden zum Zeitpunkt der Berichterstellung 98% aller Mikroprozessoren in solchen Systemen eingesetzt. In gewisser Hinsicht sind eingebettete Systeme eine Voraussetzung für viele Produkte und sie haben einen Einfluss auf das Marktvolumen aller oben genannten Bereiche. Das gesamte Marktvolumen zu beziffern ist aber sehr schwer, denn es liegt deutlich über dem Marktvolumen der eingesetzten IKT-Komponenten. Es würde in die Irre führen, wenn man sich nur auf den Marktwert der Halbleiter beziehen würde, denn deren Wert macht nur einen Bruchteil des gesamten Wertes aus.

Die ökonomische Bedeutung von CPS und IoT spiegelt sich wider in Aufrufen zur Einreichung von Projektvorschlägen von fördernden Institutionen, beispielsweise in den USA [116] und in Europa [156].

1.3 Herausforderungen

Leider geht der Entwurf von eingebetteten Systemen und deren Integration in CPS- und IoT-Systeme einher mit einer großen Zahl von schwierigen Entwurfsaspekten, wie u.a. den nachfolgend aufgeführten:

- Cyber-physikalische Systeme müssen **verlässlich** sein.

Definition 1.4: Ein System ist **verlässlich**, wenn es den vorgesehenen Dienst mit hoher Wahrscheinlichkeit bereitstellt und keinen Schaden anrichtet.

Verlässlichkeit ist erforderlich, weil diese Systeme mit der physischen Umgebung verbunden sind und einen direkten Einfluss auf diese Umgebung haben. Dieser Aspekt muss während des gesamten Entwurfsprozesses beachtet werden.

Verlässlichkeit umfasst eine Reihe von Teilaspekten:

1. **Informationssicherheit** (engl. *security*)⁴: Informationssicherheit hat den Schutz von Informationen als Ziel. Dabei können Informationen sowohl auf Papier, in Rechnern oder auch in Köpfen gespeichert sein.

⁴ Im Deutschen werden die hier angeführten Begriffe „*safety*“ und „*security*“ meist beide durch das Wort „Sicherheit“ wiedergegeben. Wir verwenden hier den Begriff „Informationssicherheit“ für *security* und „Betriebssicherheit“ für *safety*.

Definition 1.5 ([256, 75]): Die Informationssicherheit ist definiert als die Gewährleistung von Vertraulichkeit, Integrität und Verfügbarkeit.

Diese Gewährleistung kann durch Diebstahl oder Schäden aufgrund von Angriffen von außen gefährdet sein. Solche Angriffe können erfolgen, wenn CPS- oder IoT-Systeme mit der Umgebung verbunden werden. Cyber-Kriminalität und Cyber-Kriegsführung sind Beispiele solcher Angriffe mit einem Potential für erhebliche Schäden. Je mehr Komponenten man verbindet, umso mehr Angriffe und Schäden werden möglich. Dies ist zu einem schwerwiegenden Problem beim Entwurf und der Verbreitung von IoT-Systemen geworden.

Die einzig sichere Lösung ist es, die Komponenten nicht mit der äußeren Umgebung zu verbinden, was aber der ursprünglichen Intention der Nutzung widerspricht. Zur Gewährleistung der Informationssicherheit sollen nach Ravi et al. [301] (mindestens) die folgenden Anforderungen erfüllt werden:

- Ein **Benutzer-Identifikationsprozess** überprüft Identitäten, bevor Benutzer ein System verwenden dürfen.
 - Ein **sicherer Netzwerkzugang** erlaubt eine Netzwerkverbindung nur dann, wenn das kommunizierende Gerät autorisiert ist.
 - Eine **sichere Kommunikation** erfüllt dafür notwendige Eigenschaften.
 - Eine **sichere Speicherung** verlangt Vertraulichkeit und Integrität der Daten.
 - **Informationssicherheit der Inhalte** verlangt Einschränkungen hinsichtlich der Nutzung der Information.
2. **Vertraulichkeit** ist eine der Anforderungen an die Informationssicherheit. Sie kann wie folgt definiert werden:

Definition 1.6 ([256, 75]): „**Vertraulichkeit** ist der Schutz vor unbefugter Preisgabe von Informationen. Vertrauliche Daten und Informationen dürfen ausschließlich Befugten in der zulässigen Weise zugänglich sein”.

Vertraulichkeit wird üblicherweise mit Techniken zur Konstruktion informationssicherer Systeme hergestellt, z.B. durch Verschlüsselung.

3. **Betriebssicherheit (engl. *safety*):**

Definition 1.7 ([251]): **Betriebssicherheit** kann definiert werden als „Abwesenheit nicht annehmbarer Risiken einer physischen Verletzung oder eines Schadens der Gesundheit von Personen, weder direkt noch indirekt durch einen Schaden am Eigentum oder der Umgebung”.

„**Funktionelle Betriebssicherheit** ist der Teil der gesamten Betriebssicherheit, die davon abhängt, dass ein System oder Gerät auf Eingaben korrekt reagiert”.

„Im Kontext von Computersystemen wird der Begriff auch zur Abgrenzung von einer Sicherheit gegen äußere Angriffe, etwa durch Schadsoftware verwendet. Betriebssicherheit bezieht sich dann im Unterschied dazu auf Gefahren durch Ausfälle, die ohne Fremdeinwirkung auftreten, also etwa Hardware- oder Stromausfälle, fehlerhaft geschriebene Software oder Bedienungsfehler” [574].

4. **Zuverlässigkeit:** Dieser Begriff bezieht sich auf Fehlfunktionen von Systemen, die daraus resultieren, dass Komponenten nicht gemäß ihrer Spezifikation arbeiten, z.B. weil sie „kaputt“ gehen. Zuverlässigkeit kann definiert werden als die Wahrscheinlichkeit, dass ein System den erwarteten Dienst für eine gewisse Zeitdauer erbringt⁵ und sie kann als ein Aspekt der Betriebssicherheit gesehen werden. Im Rahmen der Beurteilung der Zuverlässigkeit betrachten wir keine Angriffe auf das System von außen, sondern nur Effekte, die im System während seines normalen Betriebs entstehen.
5. **Wartbarkeit:** Die Wartbarkeit ist die Wahrscheinlichkeit, dass ein ausgefallenes System innerhalb einer bestimmten Zeitspanne repariert werden kann.
6. **Verfügbarkeit:** Die Verfügbarkeit ist die Wahrscheinlichkeit, dass das System verfügbar ist. Sowohl Zuverlässigkeit wie auch Wartbarkeit müssen hoch sein, um eine hohe Verfügbarkeit zu erzielen und damit zur Verlässlichkeit beizutragen.

Entwickler eingebetteter Systeme könnten sich nun anfangs nur auf die Funktionalität eines Systems konzentrieren und versuchen, Verlässlichkeit zu einem funktionierenden Entwurf hinzuzufügen. Dieser Ansatz funktioniert in den meisten Fällen jedoch nicht, da bestimmte Entwurfsentscheidungen verhindern, dass beispielsweise die erforderliche Zuverlässigkeit im Nachhinein erreicht werden kann. Wenn die physikalische Aufteilung eines Systems falsch gewählt wurde, kann es unmöglich sein, ein redundantes System zu entwerfen. Daher darf „Zuverlässigkeit nicht erst nachträglich zu einem System hinzugefügt werden“, vielmehr muss sie von Anfang an berücksichtigt werden [304]. Es müssen gute Kompromisse gefunden werden, um ein ausreichendes Maß an Betriebssicherheit und Informationssicherheit zu finden [297].

Auch perfekt entworfene Systeme können ausfallen. Dies kann daran liegen, dass die zu Grunde liegenden Annahmen über die Auslastung und möglicherweise auftretende Fehler nicht korrekt waren [304]. So kann ein System beispielsweise ausfallen, wenn es außerhalb des ursprünglich angenommenen Temperaturbereichs betrieben wird.

- Wenn wir uns die Schnittstelle zwischen der physischen und der Cyber-Welt genauer ansehen, stellen wir fest, dass physikalische und Cyber-Modelle nicht richtig zusammenpassen. Die nachfolgende Liste zeigt Beispiele:
 - Viele CPS-Systeme müssen Zeitschranken einhalten. Die Verletzung von Zeitschranken kann zu einem ernsten Qualitätsverlust für bereitgestellte Dienste führen (wenn beispielsweise die Audio- oder die Videoqualität beeinträchtigt ist) oder dem Benutzer einen Schaden zufügen (wenn beispielsweise Autos, Eisenbahnen oder Flugzeuge nicht in der erwarteten Weise funktionieren). Einige Zeitschranken heißen harte Zeitschranken:

Definition 1.8 (Kopetz [304]): Eine Zeitschranke heißt **harte** Zeitschranke, wenn ihr Nicht-Einhalten eine Katastrophe verursachen kann. Alle anderen Zeitschranken heißen **weiche** Zeitschranken.

⁵ Eine formale Definition dieses Begriffs wird in Definition 5.36 auf Seite 307 gegeben.

Viele der aktuellen informationsverarbeitenden Systeme verwenden Techniken, um die **durchschnittliche** Geschwindigkeit der Informationsverarbeitung zu erhöhen. So verbessern beispielsweise Caches die durchschnittliche Leistung eines Systems. In anderen Fällen wird zuverlässige Kommunikation realisiert, indem bestimmte Übertragungen wiederholt werden. Im Durchschnitt haben diese Wiederholungen nur einen (hoffentlich kleinen) Verlust an Übertragungsleistung zur Folge, auch wenn die Kommunikationsverzögerung für eine bestimmte Nachricht einige Größenordnungen über der normalen Verzögerung liegen kann. Im Kontext von Echtzeitsystemen sind Diskussionen über durchschnittliche Leistung oder Verzögerung eines Systems aber nicht akzeptabel. „**Eine garantierte Systemantwort muss ohne Verwendung statistischer Argumente erklärt werden können**“ [304].

Viele Modellierungstechniken der Informatik benutzen keine echte Zeit. Häufig wird die Zeit ohne Benutzung physikalischer Einheiten modelliert, womit keine Unterscheidung zwischen Picosekunden und Jahrhunderten gemacht wird. Die resultierenden Probleme wurden von Edward Lee sehr klar benannt: „Das Fehlen der Zeit in Kernmodellen (der Informatik) ist aus Sicht der Software eingebetteter Systeme ein Fehler“ [333].

- Viele eingebettete Systeme sind **hybride Systeme**, sie beinhalten analoge und digitale Bestandteile. Analoge Komponenten verwenden kontinuierliche Signale über einem kontinuierlichen Zeitbereich, wogegen digitale Komponenten diskrete Signalwerte über einem diskreten Zeitbereich verwenden. Viele physikalische Einheiten werden von einem Paar dargestellt, bestehend aus einer reellen Zahl und einer Einheit. Die Menge der reellen Zahlen ist überabzählbar. In der Cyber-Welt ist die Menge der darstellbaren Werte endlich. Daher können fast alle physikalischen Größen auf Computern nur approximiert werden. Bei der Simulation von physikalischen Systemen auf Computern nehmen wir üblicherweise an, dass uns diese Approximation sinnvolle Ergebnisse liefert. Taha [522] hat die Konsequenzen der Nicht-Verfügbarkeit von reellen Zahlen in der Cyber-Welt beschrieben.
- Physikalische Systeme können den sogenannten **Zeno-Effekt** aufweisen. Der Zeno-Effekt kann anhand des Beispiels des springenden Balls eingeführt werden. Angenommen, wir lassen einen Ball von einer bestimmten Höhe auf den Boden fallen. Nachdem wir den Ball loslassen, wird er beginnen zu fallen und dabei durch die Erdbeschleunigung beschleunigt werden. Wenn er auf dem Boden auftrifft, wird er aufgrund der Annahme eines **teilweise elastischen Verhaltens** reflektiert werden, d.h. er bewegt sich in die umgekehrte Richtung. Wir nehmen an, dass die Reflexion eine bestimmte Dämpfung besitzt und dass der Betrag der Geschwindigkeit unmittelbar nach der Reflexion gegenüber der Geschwindigkeit unmittelbar vor der Reflexion um einen Faktor s reduziert ist. Aufgrund dieser Annahme sprechen wir von einem **teilelastischen Stoß**. s heißt **Stoßzahl**. Aufgrund der Dämpfung wird der Ball nach dem Stoß nicht wieder die anfängliche Höhe erreichen. Außerdem wird die Zeit bis zum zweiten Aufprallen auf dem Boden kürzer sein als beim ersten Mal. Dieser Vorgang wird sich wiederholen, mit kürzeren und kürzeren Ab-

ständen zwischen den Stößen. Aufgrund des idealen Modells teilelastischer Stöße wird sich dieser Vorgang aber immer wiederholen. Abb. 1.5 zeigt ein Weg-Zeit-Diagramm (die Höhe als Funktion der Zeit) des teilelastischen Balls.

Abb. 1.5 Weg-Zeit-Diagramm eines teilelastischen Balls (© Openmodelica)

Sei nun Δ ein beliebiges Zeitintervall mit einem beliebigen Anfangszeitpunkt. Gibt es eine obere Schranke für die Anzahl der Stöße in diesem Zeitintervall? Nein, eine solche Schranke gibt es nicht, denn im idealen Modell ereignen sich die Stöße immer wieder, aber die Zeitabstände werden immer kürzer. Dies ist ein spezieller Fall des Zeno-Effekts. Ein System zeigt einen **Zeno-Effekt**, wenn es möglich ist, eine unbegrenzte Anzahl von Ereignissen in einem Intervall von endlicher Länge zu haben [404]. Mathematisch gesehen ist das möglich, denn eine unendliche Reihe kann zu einem endlichen Wert konvergieren. In diesem Fall konvergiert die unendliche Zeitreihe der Zeitpunkte der Reflexionen zu einer endlichen Zeit. In Digitalrechnern können wir eine unbegrenzte Zahl von Ereignissen nur approximieren. Weitere Details zu diesem Beispiel werden ab Seite 50 diskutiert werden. Wichtig ist, dass wir in einem Digitalrechner eine unbegrenzte Zahl von Ereignissen nur approximieren können.

- Viele eingebettete Systeme enthalten Regelschleifen wie in Abb. 1.6.

Abb. 1.6 Regelschleife

Die Theorie von Regelschleifen basierte ursprünglich auf analogen, zeitkontinuierlichen Rückkopplungen. Für digitale, zeitdiskrete Rückkopplungen wurde standardmäßig über Jahrzehnte hinweg eine periodische Abtastung eingesetzt und diese funktionierte hinreichend. Dennoch ist die periodische Abtastung nicht der bestmögliche Weg. Wir können Ressourcen einsparen, indem wir die Abtastintervalle in Zeiten relativ konstanter Signale vergrößern. Dies ist die Idee der **adaptiven Abtastung**. Adaptive Abtastung ist Gegenstand aktiver Forschung [210].

- Traditionelle sequentielle Programmiersprachen sind nicht der beste Weg, nebenläufige, zeitbehaftete Systeme zu beschreiben.
- Traditionellerweise liefert die Überprüfung einer korrekten Implementierung einer Spezifikation ein Boolesches Ergebnis: entweder ist die Implementierung korrekt oder nicht. Allerdings sind zwei physikalisch vorhandene Implementierungen nie exakt identisch, sie könnten beispielsweise in ihren Abmessungen geringfügig voneinander abweichen. Dies führt zu einer Unschärfe in der Beurteilung und die Boolesche Verifikation müsste durch eine **unscharfe Verifikation** (engl. *fuzzy verification*) ersetzt werden [446, 184].
- Nach Edward Lee [332] kann die Kombination eines deterministischen physikalischen Modells und eines deterministischen Cyber-Modells möglicherweise ein **nicht-deterministisches Modell** sein. Nicht-deterministische Abtastung kann dafür eine Ursache sein.

Insgesamt beobachten wir eine Diskrepanz zwischen der physischen und der Cyber-Welt. Dementsprechend suchen wir nach geeigneten Modellen für cyberphysikalische Systeme, aber wir können nicht erwarten, dass wir diese Diskrepanz vollkommen beseitigen können.

- Eingebettete Systeme müssen **effizient mit Ressourcen umgehen**. Dazu müssen wir uns der verwendeten Ressourcen gewahr sein. Die folgenden Metriken können verwendet werden, um die Effizienz eingebetteter Systeme zu bewerten:

1. **Energie:** Die elektronische Datenverarbeitung (EDV) benötigt, wie der Name es erwarten lässt, für die Verarbeitung von Daten elektrische Energie. Die Menge der benötigten Energie wird häufig „verbrauchte Energie“ genannt. Streng genommen ist das falsch, denn die Energie bleibt insgesamt erhalten. Tatsächlich wandeln wir (nur) elektrische Energie in eine andere Form der Energie, üblicherweise in thermische Energie. Für eingebettete Systeme ist die Verfügbarkeit **elektrischer** Leistung und Energie (als Integral der Leistung über der Zeit) ein entscheidender Faktor. Dies wurde bereits im Rahmen einer niederländischen Technologievorhersage bemerkt: „(elektrische) Leistung wird als die wichtigste Randbedingung bei eingebetteten Systemen betrachtet“ [150].

Warum sollten wir uns der Menge der umgewandelten **Energie gewahr** sein, d.h. uns um diese Menge kümmern (und sie auch so sparsam wie möglich einsetzen)? Dafür gibt es viele Gründe, ein großer Teil davon ist für alle Systemklassen anwendbar, aber es gibt auch weniger relevante Kombinationen von Gründen und Systemklassen, wie in Tabelle 1.1 zu sehen ist.

Tabelle 1.1 Abhängigkeit der Relevanz von Energieeinsparungsgründen von der Systemklasse

Systemklasse	Während der Nutzung relevant?		
	Netzbetrieb	Akkubetrieb	ohne Netz
Beispiel	Fabrik	Laptop	Sensor-Netzwerk
Globale Erwärmung	ja	kaum	nein
Kosten der Energie	ja	kaum	üblicherweise nicht
Höhere Performanz	ja	ja	ja
Laufzeit ohne Netz	nein	ja	ja
Thermische Effekte, Überhitzung	ja	ja	ja
Vermeiden hoher Ströme, Metallwanderung	ja	ja	ja
Kaum Energie vorhanden	nein	selten	ja

Die globale Erwärmung ist natürlich ein sehr wichtiger Grund, um des Verbrauchs an elektrischer Energie gewahr zu sein. Allerdings ist die Menge an verfügbarer Energie bei mobilen Systemen üblicherweise sehr klein und daher ist der Beitrag dieser Systeme zur globalen Erwärmung ebenfalls klein⁶. Die Kosten der Energie sind relevant, wenn die benötigte Menge an Energie teuer ist. Bei Systemen mit Netzbetrieb kann dies aufgrund großer Mengen an Energie passieren. Für Systeme ohne Netzbetrieb sind diese Mengen meist klein, aber es kann Fälle geben, in denen selbst kleine Mengen teuer sind. Eine größere Performanz (Rechenleistung) verlangt – sofern nicht Fortschritte in der Technologie diese ermöglichen – i.d.R. nach mehr Energie. Die Laufzeit ohne Stromnetz ist z.B. für Mobiltelefone und Laptops relevant. Thermische Effekte werden wichtiger, weil zu heiße Systeme nicht benutzbar sind und weil die Zuverlässigkeit von Systemen mit steigenden Temperaturen sinkt. Es kann sogar notwendig sein, Teile von Silizium-Chips abzuschalten, d.h. von der Stromversorgung zu trennen, um eine Überhitzung zu vermeiden. Diese Notwendigkeit ist unter dem Begriff *dark silicon* bekannt geworden [153]. Daher müssen thermische Effekte ebenfalls betrachtet werden. Hohe Stromdichten können dazu führen, dass Metalle in Chips sich nicht mehr wie Festkörper verhalten, sondern wandern. In einigen Fällen, wie z.B. räumlich entfernt aufgestellten Sensorknoten, ist Energie eine wirklich sehr knappe Ressource. Es ist interessant anzusehen, in welchen Fällen ein Grund für effiziente Nutzung der Energie nicht so relevant ist. Bei Systemen, die mit dem Stromnetz verbunden sind, ist Energie i.d.R. keine knappe Ressource. Bei Systemen, die nicht mit dem Stromnetz verbunden sind, ist der Einfluss auf die globale Erwärmung gering, weil die Kapazität von Stromspeichern begrenzt ist.

⁶ Dies sei durch ein konkretes Zahlenbeispiel belegt: ein derzeit als groß empfundener Akku eines Mobiltelefons hätte beispielsweise eine Kapazität von 3600 mAh und für die Spannung nehmen wir eine mittlere Spannung von 4 V an. Damit ergibt sich eine Energie von 14,4 Wh. Eine vollständige Ladung entspricht damit dem Verbrauch eines i.d.R. permanent eingeschalteten Internet-Zugangsknotens in ca. 1-2,5 Stunden bzw. eines Fernsehgerätes im Bruchteil einer Stunde.

Die Bedeutung der Energie- und Leistungseffizienz wurde zunächst bei eingebetteten Systemen wahrgenommen und wurde später für allgemeine Rechneranwendungen zur Kenntnis genommen. Die Bedeutung der Energieeffizienz führte zu Initiativen wie der *green computing initiative* [11].

Allgemein sollte nicht vergessen werden, dass es nicht nur auf den Energieverbrauch während des Betriebes ankommt. Vielmehr hat auch die Herstellung von Systemen einen erheblichen Anteil am Energieverbrauch. Es sollte daher der komplette Lebenszyklus eines Produktes im Rahmen eines *Life Cycle Assessments* (LCA) bewertet werden [375]. Dementsprechend kann man den Einfluss auf die Umwelt reduzieren, indem man seltener ein neues Gerät kauft.

2. **Laufzeit-Effizienz:** Eingebettete Systeme sollten die verfügbare Hardwarearchitektur so gut wie möglich ausnutzen. Eine ineffiziente Nutzung von Ausführungszeit (z.B. verschwendete Prozessorzyklen) sollte vermieden werden. Dies impliziert eine Optimierung der Ausführungszeiten über alle Ebenen hinweg, von den Algorithmen bis zu Hardwareimplementierungen.
3. **Codegröße:** Bei manchen eingebetteten Systemen muss der Code für die Ausführung von Software in dem System selbst gespeichert werden. Hierfür kann es aber sehr strenge Randbedingungen geben. Dies gilt insbesondere für *Systems on a Chip* (SoCs), d.h. Systeme, bei denen die gesamte Informationsverarbeitung auf einem *Chip* integriert ist. Wenn auch der Speicher mit auf diesem *Chip* integriert ist, muss er sehr effizient genutzt werden. Beispielsweise kann es medizinische Geräte geben, die im menschlichen Körper implantiert werden. Aufgrund der Beschränkungen der Größe und der Kommunikationsmöglichkeiten müssen diese Geräte sehr kompakt sein.

Allerdings kann sich die Bedeutung dieser Einschränkung ändern, wenn das dynamische Laden von Code akzeptabel wird oder wenn größere Speicherdichten (gemessen in Bits pro Volumeneinheit) verfügbar werden. *Flash*-Speicher und neue Speichertechnologien können potentiell einen großen Einfluss haben.

4. **Gewicht:** Tragbare Systeme dürfen nicht schwer sein. Ein geringes Gewicht ist oft ein wichtiges Argument für den Kauf eines bestimmten Gerätes.
5. **Kosten:** Eingebettete Systeme für den Massenmarkt, wie Konsumelektronik, werden in hohen Stückzahlen hergestellt. Bei diesen Geräten ist die Wettbewerbsfähigkeit besonders wichtig und daher ist eine effiziente Nutzung der Hardwarekomponenten und des Budgets für Softwareentwicklung erforderlich. Zur Implementierung der benötigten Funktionalität sollte eine minimale Menge an Ressourcen verwendet werden. Wir sollten also die Anforderungen mit der geringsten Menge an Hardwareressourcen und Energie erfüllen können. Um den Energieverbrauch zu reduzieren, sollten Taktfrequenzen und Versorgungsspannungen so gering wie möglich sein. Zudem sollten nur die unbedingt notwendigen Hardwarekomponenten vorhanden sein. Komponenten, welche die maximale Ausführungszeit (engl. *Worst Case Execution Time* (WCET)) nicht verbessern (wie z.B. viele Caches oder virtuelle Speicherverwaltungseinheiten), können in manchen Fällen weggelassen werden.

Um geforderte Effizienzkriterien zu erfüllen, kann der Softwareentwurf nicht unabhängig von der darunterliegenden Hardware erfolgen. Daher müssen in allen Entwurfsschritten sowohl die Software wie auch die Hardware berücksichtigt werden. Dies ist jedoch schwierig, da solche integrierten Ansätze normalerweise nicht gelehrt werden. Die Zusammenarbeit zwischen Elektrotechnik und Informatik hat den dafür erforderlichen Umfang noch nicht erreicht.

Die beste Energieeffizienz würde bei einer Abbildung der Spezifikationen auf Hardware erzielt werden. Hardwareimplementierungen sind aber sehr teuer und erfordern eine lange Entwurfsdauer. Daher bieten Hardwareentwürfe nicht die benötigte Flexibilität, um Entwürfe bei Bedarf anzupassen. Wir benötigen also einen guten Kompromiss zwischen Effizienz und Flexibilität.

- CPS- und IoT-Systeme sammeln sehr häufig große Mengen an Daten. Diese Daten müssen gespeichert werden und sie müssen analysiert werden. Daher gibt es eine starke Verbindung zwischen der Datenanalyse bzw. Maschinellen Lernen und CPS/IoT. Dies ist genau der Gegenstand unseres Sonderforschungsbereichs SFB 876⁷. Der SFB 876 hat seinen Schwerpunkt beim Maschinellen Lernen unter Ressourcen-Beschränkungen.
- **Einfluss jenseits technischer Fragen:** Aufgrund des großen Einflusses auf die Gesellschaft müssen auch rechtliche, ökonomische, soziale, menschliche und umwelttechnische Folgen betrachtet werden:
 - Die Integration vieler Komponenten, möglicherweise von verschiedenen Herstellern, lässt die Frage nach der Haftung aufkommen. Diese Frage wird beispielsweise für autonom fahrende Autos diskutiert. Auch muss geklärt werden, wer welche Rechte hat. Es kann nicht akzeptiert werden, dass alle Rechte einer Firma gehören.
 - Die sozialen Folgen der neuen IKT-Geräte müssen auch bedacht werden. Dies hat zur Einführung des Begriffs des *Cyber Physical Social Systems* (CPSS) geführt [140]. Gegenwärtig werden diese Folgen häufig erst lange nach der Verfügbarkeit der Geräte bemerkt, wie das beispielsweise bei Mobiltelefonen der Fall ist.
 - Benutzerfreundliche Mensch-Maschine-Schnittstellen müssen verfügbar sein, um nicht größere Teile der Gesellschaft von der Anwendung auszuschließen.
 - Der Beitrag zur globalen Erwärmung und der entstehende Abfall sollte möglichst gering sein und einen akzeptablen Umfang nicht übersteigen. Dasselbe gilt für den Ressourcen-Verbrauch.
- Reale Echtzeitsysteme sind hochgradig **nebenläufig** (engl. *concurrent*). Die Verwaltung der Nebenläufigkeit ist eine weitere große Herausforderung.
- CPS- und IoT-Systeme bestehen typischerweise aus **heterogenen** Hard- und Softwarekomponenten von verschiedenen Herstellern und sie müssen in einer sich verändernden Umgebung funktionieren. Die Heterogenität führt zu Herausforderungen für die korrekte Kooperation der verschiedenen Komponenten. Es ist nicht ausreichend, nur den Software- oder den Hardwareentwurf zu betrachten. Auf-

⁷ Siehe <http://www.sfb876.tu-dortmund.de>.

grund der Entwurfskomplexität muss ein hierarchischer Ansatz gewählt werden. Reale eingebettete Systeme bestehen aus vielen Komponenten und wir interessieren uns daher für den **kompositionellen** Entwurf. Das bedeutet, wir möchten die Folgen der Integration von Komponenten analysieren [214]. Beispielsweise möchten wir wissen, ob wir ein Navigationssystem zu den Informationsquellen im Auto hinzufügen können, ohne den Kommunikationsbus zu überlasten.

- Der Entwurf von CPS-Systemen erfordert Wissen aus vielen Gebieten. Es ist schwierig, Mitarbeiter zu finden, die in allen relevanten Bereichen ein ausreichendes Wissen haben und schon der Austausch von Wissen zwischen diesen Bereichen ist eine Herausforderung. Eine noch größere Herausforderung ist es, ein Ausbildungsprogramm für den Entwurf von CPS-Systemen zu entwickeln, da es enge Obergrenzen für den studentischen Arbeitsaufwand gibt [379]. Insgesamt wäre es erforderlich, die Wände zwischen den Disziplinen und Fakultäten niederzureißen, oder sie zumindest niedriger zu gestalten.

Eine Liste der Herausforderungen findet sich auch im Bericht über IoT von Sundmaecker et al. [516].

1.4 Gemeinsame Eigenschaften

Zusätzlich zu den aufgeführten Herausforderungen gibt es weitere gemeinsame Eigenschaften von eingebetteten, cyber-physikalischen und IoT-Systemen, unabhängig vom Anwendungsbereich.

- Diese Systeme werden oft durch **Sensoren**, die Informationen über diese Umgebung sammeln, und **Aktuatoren**, welche die Umgebung steuern, mit der physischen Umgebung verbunden. Beim Internet der Dinge werden diese Komponenten mit dem Internet verbunden.

Definition 1.9: Aktuatoren sind Komponenten, die numerische Werte in physikalische Effekte verwandeln.

- Üblicherweise sind eingebettete Systeme reaktive Systeme, die wie folgt definiert werden können:

Definition 1.10 (Bergé [566]): „Ein reaktives System ist ein System, das sich in ständiger Interaktion mit seiner Umgebung befindet und mit einer Geschwindigkeit reagiert, die durch diese Umgebung vorgegeben wird.“

Reaktive Systeme modellieren wir als Systeme, die sich in einem Zustand befinden und auf eine Eingabe warten. Für jede Eingabe wird eine bestimmte Berechnung durchgeführt und eine Ausgabe sowie ein neuer Zustand erzeugt. Daher stellen endliche Automaten sehr gute Modelle für solche Systeme dar. Mathematische Funktionen zur Beschreibung der von Algorithmen zu lösenden Probleme wären hier ein unpassendes Modell.

- Eingebettete Systeme sind **in der Lehre und in der öffentlichen Diskussion unterrepräsentiert**. Reale eingebettete Systeme sind komplex. Daher wird eine umfangreiche Ausstattung benötigt, um das Thema realistisch zu unterrichten. Allerdings kann die Lehre des CPS-Entwurfs aufgrund des sichtbaren Einflusses auf das physische Verhalten auch attraktiv sein.
- Diese Systeme sind meist **für eine bestimmte Anwendung ausgelegt**. Prozessoren, die Steuerungssoftware in einem Auto oder einem Zug ausführen, werden i.d.R. nur diese Software ausführen. Es wird niemand versuchen, ein Computerspiel oder eine Tabellenkalkulation auf diesen Prozessoren auszuführen. Dafür gibt es hauptsächlich zwei Gründe:
 1. Die Ausführung zusätzlicher Programme würde diese Systeme weniger verlässlich machen.
 2. Die Ausführung zusätzlicher Programme ist nur möglich, wenn es ungenutzte Ressourcen, wie z.B. Speicher, gibt. In einem effizienten System sollten aber keine ungenutzten Ressourcen vorkommen.

Allerdings ändert sich die Situation langsam. Beispielsweise zeigt die AUTOSAR-Initiative mehr Dynamik in der Automobilindustrie [27].

- Die meisten eingebetteten Systeme verwenden weder Tastaturen, Mäuse noch große Bildschirme für ihre Benutzerschnittstellen. Stattdessen existiert eine **dedizierte Benutzerschnittstelle**, bestehend aus Schaltern, Lenkrädern, Pedalen usw. Daher nimmt der Benutzer meist nicht wahr, dass in solchen Systemen Informationen verarbeitet werden. Aus diesem Grund wird diese neue Ära der Rechner auch als Ära des **verschwindenden Rechners** bezeichnet.

Die Tabelle in 1.2 stellt einige herausragende Unterschiede zwischen dem Entwurf PC-ähnlicher Systeme und dem Entwurf eingebetteter Systeme dar, die bei der Abbildung von Anwendungen auf Hardwareplattformen relevant sind.

Tabelle 1.2 Unterschiede zwischen PC/Server-ähnlichen und eingebetteten Systemen

	Eingebettet	PC/Server-ähnlich
Architekturen	Häufig heterogen, sehr kompakt	Meist homogen, nicht kompakt (x86 usw.)
x86-Kompatibilität	Weniger wichtig	Sehr wichtig
Festgelegte Architektur?	Selten	Ja
Berechnungsmodell (MoC)	C+div. Modelle (Datenfluss, diskrete Ereignisse, ...)	Meist von Neumann (C, C++, Java)
Optimierungsziele	Mehrere (Energie, Größe, ...)	Durchschnittliche Performanz überwiegt
Sicherheitskritisch?	Möglicherweise	Kaum
Echtzeitsystem?	Häufig	Selten
Anwendungen zur Entwurfszeit bekannt	Die meisten oder alle	Nur einige (z.B. WORD)

Die Kompatibilität mit von x86-PCs her bekannten Befehlssätzen ist für eingebettete Systeme weniger wichtig, da es normalerweise möglich ist, Softwareanwendungen für eine gegebene Architektur neu zu übersetzen. Sequenzielle Programmiersprachen eignen sich nicht gut für die Aufgabe, nebenläufige Echtzeitsysteme zu beschreiben. Hier könnten andere Ansätze zur Modellierung von Anwendungen vorteilhafter sein. Während des Entwurfs eingebetteter/cyber-physikalischer Systeme müssen verschiedene Ziele berücksichtigt werden. Außer der durchschnittlichen Rechenleistung (Performanz) sind hier die größtmögliche Ausführungszeit (WCET), der Energieverbrauch, das Gewicht, die Zuverlässigkeit, die Betriebstemperaturen usw. mögliche Optimierungsziele. Das Einhalten von Echtzeitbedingungen ist für cyber-physikalische Systeme sehr wichtig, aber nur selten für PC-Systeme. Das Einhalten von Zeitbedingungen kann nur dann zur Entwurfszeit verifiziert werden, wenn alle Anwendungen auch zur Entwurfszeit bekannt sind. Zudem muss bekannt sein, welche Anwendungen gleichzeitig ablaufen sollen. So müssen Entwickler beispielsweise sicherstellen, dass eine GPS-Anwendung, ein Telefonanruf und Datenübertragungen gleichzeitig stattfinden können, ohne dass Sprachinformationen verloren gehen. Bei PC-Systemen ist ein solches Wissen so gut wie nie verfügbar und man arbeitet mit *best effort*-Ansätzen, d.h. man versucht, Software möglichst schnell ablaufen zu lassen, ohne ein bestimmtes Verhalten zu garantieren.

Warum versuchen wir, alle verschiedenen Arten eingebetteter Systeme in einem Buch zu behandeln? Der Grund dafür ist, dass die informationsverarbeitenden Komponenten in all diesen Systemen über viele gemeinsame Eigenschaften verfügen, auch wenn die Systeme selbst physisch sehr unterschiedlich sind.

Nicht jedes eingebettete System wird über alle der oben genannten Eigenschaften verfügen. Wir können den Begriff „eingebettetes System“ auch wie folgt definieren:

Definition 1.11: Informationsverarbeitende Systeme, welche die meisten der oben genannten Eigenschaften erfüllen, werden **eingebettete Systeme** genannt.

Diese Definition beinhaltet eine gewisse Unschärfe. Es scheint aber weder notwendig noch möglich, diese Unschärfe zu beseitigen.

1.5 Lehrplan-Integration von Eingebetteten Systemen, CPS und IoT

Leider werden eingebettete Systeme im *Computer Science Curriculum*, veröffentlicht im Jahr 2013 von ACM und der IEEE Computer Society [10], so gut wie nicht berücksichtigt. Die wachsende Bedeutung eingebetteter Systeme führt aber zu einem gesteigerten Bedarf an Ausbildung in diesem Bereich. Diese Ausbildung soll dabei helfen, die Beschränkungen aktuell verfügbarer Entwurfsmethoden zu überwinden. Eine Übersicht über Anforderungen und Ansätze zur Lehre im Bereich CPS wurden von den *National Academies of Sciences, Engineering and Medicine* [410] sowie von Marwedel et al. [379] sowie publiziert. Es gibt beispielsweise Bedarf an besse-

ren CPS-Modellen, Spezifikationssprachen, Entwurfstechniken für unterschiedliche Zielkriterien, Synthese- und Verifikationswerkzeugen, Echtzeitbetriebssystemen sowie anderer Systemsoftware. Dieses Buch soll helfen, wesentliche Fragestellungen zu vermitteln und ein Sprungbrett für weitere Forschung in diesem Gebiet sein.

Zusätzliche Informationen zum Buch gibt es unter folgender Adresse:

<http://ls12-www.cs.tu-dortmund.de/~marwedel/es-book/>

Diese Seite beinhaltet Verweise auf Foliensätze, Simulationswerkzeuge, Fehlerkorrekturen, Vorlesungsvideos und weiteres zugehöriges Material. Videos sind direkt verfügbar von

<https://www.youtube.com/user/cyphysystems>

Leser und Leserinnen, die Fehler entdecken oder Vorschläge zur Verbesserung des Buchs machen möchten, wenden sich bitte per E-Mail an:

peter.marwedel@tu-dortmund.de

Aufgrund der Verfügbarkeit dieses Buches und der Videos ist es zu empfehlen, Lehre nach dem *flipped classroom*-Konzept [376] zu erproben. Bei diesem Konzept werden die Funktionen von Stoffvorstellungen in der Hochschule und Vertiefungsphasen zu Hause weitgehend vertauscht: Ein erstes Kennenlernen des Stoffes findet anhand der Videos oder des Buches zu Hause statt, ein Vertiefen erfolgt in Gruppenarbeit während der Anwesenheit in der Hochschule. Auf diese Weise werden die Fähigkeiten zur Problemlösung, zum *Teamwork* und die sozialen Kompetenzen gestärkt. Auch wird die Verfügbarkeit des Internets genutzt, um Lehrmethoden für eingeschriebene Studierende zu verbessern. In Pandemiephasen, wie der von COVID-19, ist es vorteilhaft, dass Anwesenheitsphasen an der Hochschule auf das notwendige Minimum beschränkt werden können. Vom *flipped-classroom*-Konzept zu unterscheiden sind *Massive Open Online Courses* (MOOCs), die v.a. entfernt lebende Studierende ansprechen sollen. Zu lösende Aufgaben können entweder diesem Buch oder anderen Büchern (wie z.B. [593, 81, 174]) entnommen werden.

Beim *flipped classroom*-Konzept können Laborphasen vollständig genutzt werden, um praktische Erfahrungen mit CPS zu bekommen. Zu diesem Zweck sollte ein Kurs, der dieses Buch benutzt, um ein spannendes Labor ergänzt werden, in dem beispielsweise kleine Roboter (wie Lego *Mindstorms*TM) oder Mikrocontroller (wie *Raspberry Pi*, *Arduino* oder *Odroid*) benutzt werden. Für die handelsüblichen Mikrocontroller steht meist auch Lehrmaterial zur Verfügung. Eine andere Möglichkeit besteht in der Nutzung von Werkzeugen zur Simulation von endlichen Automaten. Empfohlen wird eine Ergänzung um eine Lehrveranstaltung zur Datenanalyse bzw. zum Maschinellen Lernen [205, 188, 559, 453].

Voraussetzungen

Das Buch setzt ein grundlegendes Verständnis verschiedener Bereiche voraus:

- Programmierung von Rechnern (mit Grundlagen des *Software Engineerings*) einschließlich erster Erfahrungen mit der Programmierung von Mikrocontrollern,
- Algorithmen (Graphenalgorithmen und Optimierungsalgorithmen wie *Branch-and-Bound*) einschließlich des Konzeptes der NP-Vollständigkeit,
- Rechnerorganisation, z.B. auf der Ebene des einführenden Buchs von J.L. Hennessy und D.A. Patterson [213] einschließlich endlicher Automaten,
- Grundlagen von Betriebssystemen (BS),
- Rechnernetz-Grundlagen (wichtig für das Internet der Dinge),
- grundlegende mathematische Konzepte (z.B. Tupel, Integrale, lineare Algebra),
- elektrotechnische (ET) Grundlagen sowie Digitalschaltungen wie Gatter und Register.

Diese Voraussetzungen lassen sich in verschiedene Kurse einteilen, wie in Abb. 1.7 dargestellt. Fehlende Grundkenntnisse im Bereich elektrischer Netzwerke, von

Abb. 1.7 Positionierung der einzelnen Themen dieses Buches

Operationsverstärkern, von Speicherverwaltung und von Ganzzahliger Linearer Programmierung können durch Lesen der Anhänge dieses Buches teilweise ausgeglichen werden. Kenntnisse in Statistik und Fourier-Reihen sind nützlich.

Empfohlene zusätzliche Lehrveranstaltungen

Das Buch sollte durch aufbauende Kurse ergänzt werden, die speziellere Kenntnisse in einigen der folgenden Gebiete vermitteln (siehe die untere Zeile von Abb. 1.7)⁸:

- Regelungstechnik,
- Digitale Signalverarbeitung,
- Bilderkennung,

⁸ Die Aufteilung in grundlegende und weiterführende Lehrveranstaltungen kann sich dabei zwischen einzelnen Universitäten unterscheiden.

- Echtzeitsysteme, Echtzeitbetriebssysteme und *Scheduling*,
- Robotik,
- Anwendungsgebiete wie Telekommunikation, Kraftfahrzeuge, medizinische Geräte und Gebäudeautomatisierung,
- *Middleware*,
- Spezifikationssprachen und -modelle für eingebettete Systeme,
- Sensoren und Aktuatoren,
- Verlässlichkeit von Rechnersystemen,
- energiesparende Entwurfstechniken,
- physikalische Aspekte von CPS,
- rechnergestützte Entwurfswerkzeuge für anwendungsspezifische Hardware,
- formale Verifikation von Hardwaresystemen,
- Test von Hardware- und Softwaresystemen,
- Leistungsbewertung von Rechensystemen,
- *ubiquitous computing*,
- Kommunikationstechniken für das Internet der Dinge (IoT),
- gesellschaftliche Bedeutung und Wirkung eingebetteter, cyber-physikalischer und IoT-Systeme sowie
- rechtliche Aspekte derselben Systeme.

1.6 Entwurfsflüsse

Der Entwurf der betrachteten Systeme stellt eine recht komplexe Aufgabe dar, die in eine Reihe von Unteraufgaben aufgeteilt werden muss. Diese Unteraufgaben müssen nacheinander und einige müssen mehrfach ausgeführt werden.

Der Entwurfsinformationsfluss beginnt mit Ideen im Kopf von Personen. Diese Ideen sollten bereits Wissen über den Anwendungsbereich beinhalten. Die Ideen müssen in einer Entwurfsspezifikation festgehalten werden. Zudem sind üblicherweise Hardware- und Systemsoftwarekomponenten vorhanden, diese sollten möglichst wiederverwendet werden (siehe Abb. 1.8). Bei diesem Diagramm (wie auch in weite-

Abb. 1.8 Vereinfachter Entwurfsfluss

ren ähnlichen Diagrammen in diesem Buch) verwenden wir **Rechtecke mit abgerundeten Ecken zur Darstellung gespeicherter Informationen** und **herkömmliche Rechtecke für Transformationen von Informationen**. Insbesondere kann die Speicherung zentral in einer **Entwurfsdatenbank** erfolgen. Diese Datenbank ermöglicht es, einen Überblick über die Entwurfsmodelle zu behalten. Normalerweise sollten die Entwurfsdaten einer Versionsverwaltung unterliegen, die z.B. mittels der Werkzeuge CVS [87], SVN [108] oder git (siehe <https://www.git-scm.com>) verfügbar ist. Eine gute Entwurfsdatenbank sollte auch eine Schnittstelle zur Entwurfsverwaltung besitzen, die es erlaubt, einen Überblick über die Anwendbarkeit von Entwurfswerkzeugen und -folgen zu erhalten. Diese Schnittstelle kann als graphische Benutzerschnittstelle (engl. *Graphical User Interface* (GUI)) realisiert werden, um die Benutzung zu vereinfachen. Die Entwurfsdatenbank und die Benutzerschnittstelle können zu einer **integrierten Entwicklungsumgebung** (engl. *Integrated Development Environment* (IDE)) zusammengefasst werden, die auch *design framework* genannt wird (siehe z.B. [346]). Eine integrierte Entwicklungsumgebung verwaltet die Abhängigkeiten zwischen den Werkzeugen und der Entwurfsinformation.

Die Verwendung der Datenbank ermöglicht es, Entwurfsentscheidungen in einem iterativen Ablauf zu treffen. Für jeden Schritt muss die Information über das Entwurfsmodell abgerufen und anschließend bewertet werden.

In den einzelnen Entwurfsdurchläufen werden **Anwendungen** auf Ausführungsplattformen **abgebildet**, zudem werden neue (partielle) Entwurfsinformationen erzeugt. Dies beinhaltet die Abbildung von Operationen auf nebenläufige Tasks, die Abbildung von Operationen auf Hardware oder Software (Hardware/Software-Partitionierung genannt), die Übersetzung und das *Scheduling*.

Entwürfe sollten in Hinblick auf verschiedene Ziele wie Rechenleistung (Performanz), Verlässlichkeit, Energieverbrauch, thermisches Verhalten usw. **bewertet (evaluiert)** werden. Der aktuelle Stand der Technik garantiert nicht, dass irgendeiner dieser Entwurfsschritte korrekt ist. Daher ist es auch erforderlich, einen Entwurf zu **validieren**. Die Validierung umfasst den Abgleich von Zwischenversionen und der endgültigen Version von Entwurfsbeschreibungen mit anderen Beschreibungen. Jeder neue Entwurf sollte bewertet und validiert werden.

Da die Effizienz eingebetteter Systeme besonders wichtig ist, sind **Optimierungen** erforderlich. Dabei ist eine Vielzahl von Optimierungen verfügbar, beispielsweise *High-Level-Transformationen* (wie z.B. erweiterte Schleifentransformationen) und Energieoptimierungen.

Wenn Überlegungen zur Testbarkeit schon in frühen Entwurfsphasen berücksichtigt werden sollen, müssen Tests in die dargestellten Entwurfsiterationen integriert werden. In Abb. 1.8 wurde die Testerzeugung als optionaler Schritt während der Entwurfsiterationen hinzugefügt (dargestellt durch das gestrichelte Rechteck). Wenn die Testerzeugung nicht Bestandteil der einzelnen Entwurfsiterationen ist, dann muss sie nach Fertigstellung des Entwurfs durchgeführt werden. Am Ende jedes einzelnen Schrittes sollte die Datenbank aktualisiert werden.

Die Details des Informationsflusses zwischen Datenbank, Anwendungsabbildung, Bewertung, Validierung, Optimierung, Testbarkeitsüberlegungen und der Speicherung von Entwurfsinformationen können durchaus abweichen. Abhängig von der

konkret verwendeten Entwurfsmethode können diese Aktionen auf vielfältige Weise miteinander verknüpft sein. Dieses Buch gibt einen breiten Überblick über den Entwurf eingebetteter Systeme und ist nicht auf bestimmte Entwurfsflüsse und Werkzeuge festgelegt. Daher haben wir keine feste Liste von Entwurfsschritten vorgegeben. Wir können die in Abb. 1.8 enthaltenen Schleifen für jede Entwurfsumgebung „abrollen“ und einzelnen Entwurfsschritten Namen zuordnen.

Dies führt bei Einsatz von SpecC [173] zu dem in Abb. 1.9 dargestellten Entwurfsfluss. Der gezeigte Fall beinhaltet eine bestimmte Menge an Entwurfsschritten, wie

Abb. 1.9 Entwurfsfluss für die SpecC-Werkzeuge (vereinfacht)

Architekturerkundung, Kommunikationssynthese und Software- wie Hardwareübersetzung. Die genaue Bedeutung dieser Begriffe ist für dieses Buch nicht relevant. In Abb. 1.9 sind Validierung und Bewertung für jeden der Schritte explizit als Bestandteil eines größeren Schrittes dargestellt.

Eine weitere Instanz von Abb. 1.8 ist in Abb. 1.10 dargestellt. Dies ist der Entwurfsfluss des V-Modells [550], der für viele deutsche IT-Projekte, speziell im öffentlichen Sektor und darüber hinaus, verbindlich ist. Abb. 1.10 zeigt die durch-

Abb. 1.10 Entwurfsfluss für das V-Modell

zuführenden Schritte im Einzelnen. Diese Schritte entsprechen bestimmten Phasen des Software-Entwicklungsprozesses (auch hier ist die genaue Bedeutung im Rah-

men dieses Buches nicht relevant). Dieses Diagramm fasst Entwurfsentscheidungen, Bewertung und Validierung in einem einzelnen Schritt zusammen. Anwendungswissen, Systemsoftware und Systemhardware werden nicht explizit dargestellt. Das V-Modell beinhaltet auch ein Modell der Integrations- und Testphase (rechter Zweig des Diagramms). Damit wird das Testen mit in die Integrationsphase einbezogen. Das dargestellte Modell entspricht dem V-Modell Version „97“. Das aktuellere V-Modell XT ermöglicht eine verallgemeinerte Menge an Entwurfsschritten. Diese Erweiterung passt sehr gut zu unserer Interpretation von Entwurfsflüssen in Abb. 1.8. Andere iterative Ansätze sind das **Wasserfallmodell** und das **Spiralmodell**. Weitere Informationen zum *Software-Engineering* für eingebettete Systeme finden sich in dem Buch von J. Cooling [109].

Unser allgemeiner Entwurfsfluss passt auch zu Flussmodellen, die im Hardwareentwurf zum Einsatz kommen. Ein weit verbreitetes Modell ist das Y-Diagramm (engl. *Y-chart*) von Gajski und Kuhn [171] (siehe Abb. 1.11). Die Autoren un-

Abb. 1.11 Gajskis Y-Diagramm und Entwurfspfad (rot, fett)

terscheiden drei Dimensionen von Entwurfsinformationen: Verhalten, Struktur und *Layout*. Die erste Dimension stellt ausschließlich das Verhalten dar. Ein Modell auf hoher Abstraktionsebene würde lediglich das Gesamtverhalten beschreiben, wogegen detailliertere Modelle das Verhalten einzelner Komponenten beschreiben würden. Modelle der zweiten Dimension beinhalten Strukturinformationen wie z.B. Informationen über Hardwarekomponenten. In dieser Dimension könnten Beschreibungen auf hoher Abstraktionsebene Prozessoren entsprechen, Beschreibungen auf unterster Ebene entsprächen Transistoren. Die dritte Dimension stellt die geometrische *Layout*-Information eines *Chips* dar. Entwurfspfade beginnen meist mit einer grobgranularen Verhaltensbeschreibung und enden mit einer feingranularen geometrischen Beschreibung. Entlang dieses Pfades entspricht jeder Schritt einer Iteration unseres generischen Entwurfsflussmodells. Im Beispiel von Abb. 1.11 erfolgt zu Beginn eine Verfeinerung im Verhaltensbereich. Der zweite Entwurfsschritt bildet das Verhalten auf Strukturelemente ab usw. Schließlich entsteht eine detaillierte geometrische Beschreibung des *Chip-Layouts*.

Diese drei Beispiele zeigen, dass der iterative Fluss von Abb. 1.8 in einer Reihe von Entwurfsflüssen verwendet wird. Die Beschaffenheit der einzelnen Iterationen aus

Abb. 1.8 wird dabei oft unterschiedlich betrachtet. Idealerweise würden wir zunächst die Eigenschaften unseres Systems beschreiben wollen und dann den Rest von einem geeigneten leistungsfähigen Werkzeug erledigen lassen. Eine solche automatische Erzeugung von Entwurfsdetails wird **Synthese** genannt.

Definition 1.12 (Marwedel [371]): „**Synthese** ist der Vorgang, die Beschreibung eines Systems in Form von Komponenten auf einer niederen Abstraktionsebene aus einer Beschreibung des erwarteten Verhaltens auf einer höheren Ebene erzeugen zu lassen.“

Bei einer automatischen Synthese soll dieser Vorgang automatisch ablaufen. Manuelle Entwurfsschritte können vermieden werden, wenn automatische Synthese erfolgreich ist. Das Ziel der automatischen Synthese wurde von Gajski [172] im „*describe-and-synthesize*“-Paradigma verfolgt. Dieses Paradigma steht im Gegensatz zu dem weiter verbreiteten „*specify-explore-refine*“-Paradigma, auch bekannt als „entwerfe und simuliere“. Der zweite Begriff betont, dass der manuelle Entwurf meist mit Simulation kombiniert werden muss, um beispielsweise Entwurfsfehler zu finden. Simulation ist hier wichtiger als bei der automatischen Synthese.

1.7 Struktur dieses Buches

Die Struktur dieses Buches folgt dem oben gezeigten Entwurfsfluss. Kapitel 2 gibt einen Überblick über Spezifikationstechniken, -sprachen und -modelle. Wichtige Hardwarekomponenten eingebetteter Systeme und die Schnittstelle zwischen der physischen Umgebung und der Informationsverarbeitung werden im Kapitel 3 vorgestellt. Kapitel 4 behandelt Systemsoftwarekomponenten, insbesondere eingebettete Betriebssysteme. Kapitel 5 beschreibt die Grundzüge der Bewertung und Validierung eingebetteter Systeme. Die Abbildung von Anwendungen auf Ausführungsplattformen ist einer der wichtigsten Schritte im Entwurfsprozess eingebetteter Systeme. Dabei ist das *Scheduling* ein wesentlicher Teil einer solchen Abbildung. Dafür werden Standard-Techniken im Kapitel 6 beschrieben. Um effiziente Entwürfe zu erzeugen, werden viele Optimierungstechniken benötigt. Eine Auswahl von Techniken aus der großen Menge verfügbarer Optimierungen wird im Kapitel 7 vorgestellt. Kapitel 8 enthält eine kurze Einführung in den Test kombinierter Hardware/Software-Systeme. Der Anhang beinhaltet eine Beschreibung einer Standard-Optimierungstechnik, einige Grundlagen zum Verständnis einer der Schaltungen aus Kapitel 3 und Grundwissen zur Speicherorganisation.

In einigen Fällen kann es erforderlich sein, spezielle Hardware für eine bestimmte Anwendung zu entwerfen oder eine Prozessorarchitektur entsprechend zu optimieren. Dieses Buch befasst sich jedoch nicht mit dem Hardwareentwurf. Coussy and Morawiec [113] geben einen Überblick über *High-Level-Hardwaresynthese-Techniken*.

Der Inhalt dieses Buchs unterscheidet sich vom Inhalt der meisten anderen Bücher zum Entwurf eingebetteter oder cyber-physikalischer Systeme. Der Schwer-

punkt vieler solcher Bücher lag früher auf der Verwendung von Mikrocontrollern, deren Speicher, Ein/Ausgabegeräten und der *Interrupt*-Struktur. Es gibt viele derartige Bücher [176, 37, 175, 177, 280, 318, 426]. Wir sind der Auffassung, dass es die steigende Komplexität eingebetteter Systeme erfordert, diesen Schwerpunkt so zu erweitern, dass zumindest unterschiedliche Spezifikationsansätze, Grundlagen von Hardwarekomponenten, die Abbildung von Anwendungen auf Ausführungsplattformen sowie die Bewertung, Validierung und Optimierungstechniken vorgestellt werden. Im vorliegenden Buch betrachten wir alle diese Bereiche. Das Ziel ist es, Studierenden eine Einführung zu eingebetteten und cyber-physikalischen Systemen zu geben, die sie in die Lage versetzt, die verschiedenen Themenbereiche einschätzen zu können.

Für weiterführende Informationen möchten wir auf eine Reihe von Quellen verweisen, von denen einige auch bei der Vorbereitung dieses Buches zum Einsatz kamen:

- Symposien mit Schwerpunkt eingebettete/cyber-physikalische Systeme sind die *Embedded Systems Week* (siehe <http://www.esweek.org>) und die *Cyber-Physical Systems Week* (siehe <http://www.cpsweek.org>).
- Die Webseite der amerikanischen virtuellen CPS-Organisation liefert zahlreiche Hinweise auf laufende Arbeiten und deren Ergebnisse [115].
- Die Webseiten einer Interessengruppe der ACM [9] befassen sich mit eingebetteten Systemen.
- Die Webseiten des europäischen Exzellenznetzwerks für eingebettete Systeme und Echtzeitsysteme [24] beinhalten viele Links zum Thema.
- Ein Buch von Edward Lee et al. befasst sich auch mit physikalischen Aspekten von cyber-physikalischen Systemen [335].
- Die Workshopreihe *Workshop on Embedded and Cyber-Physical Systems Education* (WESE) behandelt aktuelle Ansätze für die Lehre zu eingebetteten Systemen; Ergebnisse des Workshops im Jahr 2018 sind in [89] zu finden. Die Lehre ist auch Gegenstand des ersten und einzigen Workshops über Lehre im Bereich CPS [425].
- Andere Informationsquellen zu eingebetteten Systemen sind Bücher von Laplante [323], Vahid [552], die ARTIST Roadmap [63], das „*Embedded Systems Handbook*“ [614] sowie Bücher von Gajski et al. [174] und Popovici et al. [457].
- Eine große Anzahl von Quellen befasst sich mit Spezifikationssprachen. Zu diesen zählen ältere Bücher von Young [609], Burns and Wellings [79], Bergé [566] und de Micheli [124]. Weiterhin ist eine große Menge an Informationen über neue Sprachen wie SystemC [408], SpecC [173] und Java [573, 131, 71] verfügbar.
- Echtzeit-*Scheduling* wird umfassend in den Büchern von Buttazzo [81], von Krishna und Shin [311] sowie von Baruah et al. [38] behandelt.
- Ansätze für den Entwurf und die Verwendung von Echtzeit-Betriebssystemen (RTOS) werden in einem Buch von Kopetz vorgestellt [304].
- Robotik ist ein eng mit eingebetteten und cyber-physikalischen Systemen verbundenes Gebiet. Wir empfehlen das Buch von Siciliano et al. [487].
- Es gibt Bücher und Artikel, die auf das Internet der Dinge fokussiert sind [185, 193, 192].

- Sprachen und Verifikation sind der Schwerpunkt eines Buches von Haubelt und Teich [207].

1.8 Aufgaben

Die folgenden Aufgaben sollten entweder zu Hause oder während einer Anwesenheitsphase nach dem *flipped classroom*-Konzept [376] bearbeitet werden:

1.1: Nennen Sie einige der Definitionen des Begriffs „eingebettetes System“!

1.2: Definieren Sie den Begriff „cyber-physikalisches System“! Sehen Sie einen Unterschied zwischen den Begriffen „eingebettetes System“ und „cyber-physikalisches System“?

1.3: Was ist das Internet der Dinge (IoT)?

1.4: Was ist das Ziel der Initiative „Industrie 4.0“?

1.5: Auf welche Weise deckt dieses Buch den Entwurf von CPS- und IoT-Systemen ab?

1.6: In welchen Anwendungsbereichen sehen Sie Potential für den Einsatz von CPS- und IoT-Systemen? Wo erwarten Sie große Veränderungen durch die IT?

1.7: Warum sind eingebettete Systeme und cyber-physikalische Systeme wichtig?

1.8: Welche Herausforderungen müssen wir bewältigen, wenn wir das Potential von eingebetteten und cyber-physikalischen Systemen ausnutzen wollen?

1.9: Was ist eine harte Zeitschranke? Was ist eine weiche Zeitschranke?

1.10: Was ist der Zeno-Effekt?

1.11: Was ist adaptives Abtasten?

1.12: Nach welchen Kriterien können wir eingebettete Systeme bewerten?

1.13: Aus welchen Gründen und für welche Systeme betrachten wir die Energieeffizienz?

1.14: Was sind die Hauptunterschiede zwischen Anwendungen auf PCs und Anwendungen in eingebetteten Systemen?

1.15: Was ist ein reaktives System?

1.16: Auf welcher Webseite gibt es ergänzendes Material zu diesem Buch?

1.17: Vergleichen Sie das Curriculum an Ihrer Hochschule mit dem Programm, das hier in der Einleitung angegeben ist! Welche Voraussetzungen fehlen an Ihrer Hochschule? Welche fortgeschrittenen Kurse sind verfügbar?

1.18: Was ist das *flipped classroom*-Konzept?

1.19: Wie modellieren wir Entwurfsflüsse?

1.20: Was ist das V-Modell?

1.21: Wie definieren Sie „Synthese“?

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.