

Methodik und Werkzeuge zur Entwicklung optimierter Echtzeitsysteme im Kontext von SDL/MSD und VHDL

U. Herzog, W. Dulz, R. Hofmann, A. Mitschele-Thiel

Universität Erlangen, Informatik 7

Martensstr. 3, 91058 Erlangen

E-Mail: i7rp@immd7.informatik.uni-erlangen.de

Zusammenfassung

Formale Spezifikationstechniken und darauf aufbauende Werkzeuge unterstützen die systematische Entwicklung komplexer Systeme. Meist werden nur funktionale Eigenschaften überprüft; wesentliche Entwurfsvorgaben wie Leistungs- oder Echtzeitcharakteristika bleiben daher unberücksichtigt oder werden isoliert behandelt ohne in den gesamten Entwicklungsprozeß integriert zu sein. Unter Einbeziehung unserer erzielten Forschungsergebnisse auf den Gebieten der formalen Spezifikation, der Optimierung, dem Monitoring und der Leistungsbewertung von Kommunikationssystemen und Multiprozessorarchitekturen schlagen wir daher einen integrierten Gesamtansatz zur Behandlung von funktionalen als auch von echtzeitbezogenen Anforderungen vor.

Gegenstand dieses von der DFG bewilligten Projektes im Rahmen des Schwerpunktprogramms „Rapid Prototyping für integrierte Steuerungssysteme mit harten Zeitbedingungen“ ist die Entwicklung einer durchgängigen Methodik und deren Umsetzung durch ein entsprechendes Rapid-Prototyping-System zur Erstellung integrierter HW/SW-Systeme mit harten Echtzeitanforderungen sowie dessen exemplarische Anwendung zur Realisierung einer Kommunikationsanschaltung für die Übertragung von zeitkritischen Multimediadaten. Wesentliche Eckpunkte des Projektes sind 1) die formale Spezifikation des funktionalen Verhaltens als auch der Echtzeitanforderungen auf der Basis von annotierten SDL/MSD-Spezifikationen, 2) die Berechnung der Entwurfsentscheidungen (HW/SW-Partitionierung, Schedule, etc.) basierend auf den annotierten Spezifikationen und einer Komponentenbibliothek und 3) eine durchgängige spezifikationsbasierte Meß- und Bewertungsmethodik zur Validierung der Entwurfsentscheidungen und der Back-Annotation der Komponentenbibliothek.

1 Einleitung

Integrierte Steuerungssysteme haben die Realisierung einer genau definierten Anwendung zum Ziel. Oft spielen dabei Echtzeitanforderungen (harte und weiche Zeitbedingungen) eine herausragende Rolle. Die Garantie der Echtzeitbedingungen bei geringen Kosten für das System wird meist durch eine präzise Abstimmung der Komponenten des Systems erreicht, d.h. durch gegenseitige Abstimmung der Hardwarekomponenten, der Anwendungssoftware und der Systemsoftware. Der Ent-

wurf und die Realisierung von Echtzeitsystemen erfolgt in der Praxis oft ad hoc, ohne den Einsatz dafür geeigneter Werkzeuge. Bedingt durch die fortschreitende Standardisierung und die Öffnung der Märkte wird im industriellen Bereich die frühzeitige bzw. schnelle Fertigstellung neuer Produkte oder Produktvarianten immer entscheidender. Um dies zu gewährleisten, sind leistungsfähige Entwurfsmethodiken und Werkzeuge unabdingbar.

Seit Jahren gibt es eine Reihe weitverbreiteter Spezifikationstechniken, Verfahren und Werk-

zeuge, die den systematischen Entwurf technischer Produkte unterstützen; meist werden jedoch nur einzelne Entwicklungsphasen abgedeckt, oft werden wesentliche Entwurfsvorgaben nicht berücksichtigt. Darüber hinaus wird fast nie eine durchgängige Optimierung durchgeführt, auch der Einsatz meßtechnischer Methoden zur Produktentwicklung und Qualitätssicherung ist die Ausnahme.

Im folgenden geben wir einen Überblick über den Stand der Forschung auf dem Gebiet der HW/SW-Systementwicklung, charakterisieren den von uns ausgewählten Anwendungsbe- reich der Echtzeitprotokolle und geben einen Überblick über die von uns bereits erarbeiteten Erkenntnisse und Lösungsansätze.

2 Methoden und Werkzeuge für die HW/SW-Systementwicklung

2.1 Formale Beschreibungstechniken

Rapid Prototyping für den integrierten HW/SW-Entwurf reaktiver Echtzeitsysteme erfordert den Einsatz aufeinander abgestimmter CASE-Werkzeuge (CASE: Computer Aided System Engineering) unter Berücksichtigung eines übergreifenden Gesamtansatzes zur Realisierung von dedizierten HW/SW-Komponenten.

Formale Beschreibungstechniken (FDTs) dienen zur eindeutigen Festlegung insbesondere funktionaler Eigenschaften und ermöglichen den Nachweis notwendiger Korrektheitsanforderungen, die der Prototyp zu erfüllen hat. Formale Spezifikationstechniken, die im Bereich der Entwicklung von reaktiven Echtzeitsystemen starke Verbreitung gefunden haben, sind:

- *MSC*: „Message Sequence Charts“ (ITU-Standard Z.120) werden während der Anforderungsanalyse erstellt und spezifizieren das dynamische Verhalten des Systems. MSCs dienen vor allem der Formalisierung von Anforderungsdefinitionen an ein zu realisierendes System. MSCs werden meist zusammen mit SDL eingesetzt, und dienen zur automatischen Generierung von SDL-Skeletten sowie der Auswahl und Spezifikation von Testszenarien.

- *SDL*: Die „Specification and Description Language“ (ITU-Standard Z.100) dient zum Systementwurf und spezifiziert sowohl die statische Systemarchitektur als auch das funktionale Verhalten dynamischer Systeme. SDL wird in großem Umfang im Bereich der Kommunikationstechnik eingesetzt, in jüngerer Zeit aber auch sehr erfolgreich beim Entwurf anderer komplexer Echtzeitsysteme.
- *VHDL*: Die hardwarenahe „Very High Speed Integrated Circuit Hardware Description Language“ (ANSI/IEEE-Standard 1076-1993) ermöglicht sowohl die Spezifikation von HW-Systemen als auch deren Simulation und Realisierung mit Hilfe von Synthesewerkzeugen. VHDL wird eingesetzt, um bei der Entwicklung von hochintegrierten Digitalschaltungen für FPGAs und ASICs die Entwicklungszeit zu reduzieren und gleichzeitig die Entwurfsqualität zu steigern.

Für die obigen Standards existiert eine Reihe kommerzieller CASE-Werkzeuge, wie z.B. SDT [9] oder GEODE [11] zur teilautomatischen Implementierung von Echtzeit-Software aus SDL/MSC-Spezifikationen und Visual-HDL zur Generierung von Hardwarebausteinen aus VHDL-Beschreibungen. Diese Entwicklungs- umgebungen verfügen über eine Vielzahl von Teilfunktionen, wie z.B. grafisches Editieren, funktionale Simulation, Validierung der funktionalen Korrektheit oder automatische Generierung von Objektcode und HW-Bausteinen durch Compiler für Hardware und Software.

Die Erstellung hochperformanter und echtzeitfähiger Prototypen macht es erforderlich, daß neben funktionalen Aspekten auch Zeitaspekte quantitativ berücksichtigt werden müssen.

Dazu gehören vor allem

- **Zeitvorgaben**: Garantiertes Reagieren des Systems in vorgebbaren Zeitspannen oder -schranken.
- **Leistungsvorgaben**: Garantie eines gewünschten Antwortzeit/Durchsatz-Verhaltens bei vorgegebenen Normal- oder Überlastsituationen.
- **Ressourcenvorgaben**: Einschränkung der möglichen HW/SW-Lösungen.

- **Kostenvorgaben:** Berücksichtigung der Systemkosten.

Einen guten Überblick zum Stand der Technik in diesem Bereich vermitteln die Schriften und technischen Berichte des RACE-Projektes TOPIC (Tool Set for Protocol and Advanced Service Verification in IBC Environments) [10] und des ESPRIT-Projekts INSYDE (Integrated Methods for Evolving System Design) [8].

Die Sichtung der Literatur verdeutlicht, daß sich die Forschung hinsichtlich der Integration von Echtzeitzusicherungen und Leistungsbeurteilung formal spezifizierter Systeme erst in den Anfängen befindet und diesbezügliche Methoden dringend zu erforschen sind.

2.2 Entwurf und Optimierung

Zur Erstellung der Anwendungsspezifikation und der Überprüfung qualitativer Aspekte der Spezifikation, insbesondere der funktionalen Korrektheit, stehen zahlreiche Werkzeuge zur Validierung, Verifikation und zur Simulation zur Verfügung (siehe oben). Demgegenüber werden quantitative Aspekte meist erst in einem sehr späten Stadium der Realisierung berücksichtigt.

Neuere Forschungen beschäftigen sich deshalb mit der Erarbeitung von Entwurfsmethoden für die Integration quantitativer Aspekte und deren Umsetzung in Entwicklungsumgebungen. So wurden in den letzten Jahren mehrere Entwicklungsumgebungen für reine SW-Systeme, insbesondere für dedizierte parallele Rechensysteme entwickelt. Eine Übersicht über Entwicklungsumgebungen für reine SW-Systeme findet sich in [5].

Verglichen mit reinem Software- bzw. reinem Hardwareentwurf befinden sich Forschungen auf dem Gebiet des integrierten HW/SW-Entwurfs (HW/SW-Codesign) noch in den Anfängen. Eine gemeinsame Spezifikation der Anwendung unabhängig davon, welche Teile in Hardware und welche in Software realisiert werden sollen, und die späte Partitionierung in Hardware- und Softwarekomponenten bietet wesentliche Vorteile, nämlich erweiterte Möglichkeiten der Validierung bzw. Verifikation des Gesamtsystems und eine hohe Flexi-

bilität des Ansatzes gegenüber Änderungen der Anforderungen während des Entwicklungsprozesses. Zwar existieren methodische Ansätze für die Entwicklung solcher Systeme; von einer durchgehenden integrierten Methodik für Echtzeitsysteme kann aber nicht die Rede sein.

Bisherige Ansätze basieren meist auf sehr einfachen Rapid-Prototyping-Systemen (RP-System), insbesondere was die Softwarekomponenten betrifft. So verwenden die Systeme in aller Regel nur einen einzigen Prozessor bzw. Microcontroller, der die alleinige Kontrolle über das System übernimmt. Parallele Komponenten sind fast immer relativ fest und synchron mit dem Prozessor gekoppelt, was den Entwurf wesentlich erleichtert, der Leistung des RP-Systems aber enge Schranken setzt.

Ein wichtiger Aspekt des HW/SW-Entwurfs ist die Partitionierung der Spezifikation in Teile, die in Hardware und in Software realisiert werden, so daß gegebene Randbedingungen eingehalten und eine Kostenfunktion optimiert wird. Zur Bestimmung der Partitionierung existieren erste Ansätze, z.B. [4].

Bei reinen Hardwaresystemen lassen sich Echtzeitbedingungen, bedingt durch den synchronen, deterministischen Charakter der Bearbeitung und die eindeutige zeitliche und räumliche Zuordnung der Aufgabe auf die Hardwarekomponenten, relativ leicht abschätzen.

Dies ist bei Softwaresystemen, insbesondere bei Multiprozessorsystemen, bedingt durch den weniger dedizierten Charakter der Prozessoren und der asynchronen Bearbeitung, wesentlich komplexer. Insgesamt läßt sich feststellen, daß die Komplexität des Entwurfs von gemischten HW/SW-Systemen mit der Einführung stochastischer Aspekte beträchtlich ansteigt.

2.3 Implementierung

Für die Generierung von Code aus SDL-Spezifikationen existiert eine Reihe von Werkzeugen. Die kommerziellen CASE-Werkzeuge SDT und GEODE unterstützen die teilautomatische Erzeugung von C-Code. Allerdings erzeugen Codegeneratoren, die den vollen Sprachumfang von SDL unterstützen relativ ineffizienten Code.

Dies betrifft sowohl die Ausführungszeit als auch den Speicherplatzbedarf. Zunehmend kommen aber auch Codegeneratoren für Mikrocontroller auf den Markt (SDT und GEODE), die zwar nicht den vollen Sprachumfang von SDL unterstützen, dafür aber wesentlich effizienteren Code generieren.

Neben kommerziellen Werkzeugen zur Codeerzeugung steht eine Reihe von Werkzeugen aus dem universitären Bereich zur Verfügung. Zu nennen sind insbesondere das Open-SITE Projekt, das die Generierung von C++-Code unterstützt, und PAR-SDL, das die Erzeugung parallelen C-Codes für Transputernetzwerke unterstützt.

2.4 Messung und Bewertung

Bei der Messung und Bewertung von HW/SW-Systemen unterscheiden wir zwei Zielsetzungen

- die Leistungsmessung, d.h. das Messen charakteristischer Leistungsgrößen wie z.B. Durchsatz, Wartezeiten, Reaktionszeiten, und die Beobachtung des dynamischen Ablaufgeschehens, d.h. die Untersuchung des Zusammenspiels der einzelnen Systemkomponenten, um eine Begründung für gemessene Leistung zu finden.
Eingesetzt werden, je nach Aufgabenstellung unterschiedliche Monitoring-Methoden, insbesondere Software-, Hardware- und Hybridmonitoring. Eine umfassende Übersicht zu den Vor- und Nachteilen sowie ein Vergleich unterschiedlicher Realisierungen findet man in [12].
- Bei parallelen und verteilten Systemen können bereits kleine Zeitunterschiede bei der Abarbeitung paralleler Teilaufgaben oder von Kommunikationsanweisungen zu völlig unterschiedlichen Programmabläufen führen. Ein homogenes Messen, wie mit Hardware- und Hybridmonitoren möglich, ist deshalb notwendig; sehr wichtig zur Analyse und Aufbereitung der erzielten Meßergebnisse ist eine umfangreiche Auswertesoftware.

3 Lösungsansätze

3.1 Systementwicklung

Im Rahmen des DFG-Sonderforschungsbereiches 182 „Multiprozessor- und Netzwerkkonfigurationen“ wurde in den Teilprojekten B3 und C1 eine globale Methodik entwickelt, die einen integrierten Einsatz von Methoden zur Leistungsbewertung für den SDL/MSC-Entwicklungszyklus beschreibt[3]. Die wesentlichen Merkmale der Methodik sind:

- Anforderungsdefinitionen und Lastmodellierung der spezifizierten Systeme basieren auf annotierten MSC-Szenarien.
- Aktive und passive Systemkomponenten wie Prozessoren, Kanäle, Puffer oder Prozesse werden mit annotierten SDL-Spezifikationen beschrieben.
- Die explizite Einschränkung des möglichen und sehr komplexen Verhaltens des SDL-Systems auf ausgewählte annotierte MSC-Szenarien definiert einfachere Systemmodelle, die sich bezüglich Leistungs- und Echtzeitverhalten leichter beschreiben lassen. Je nach Komplexität erfolgt die Bewertung analytisch oder simulativ.
- Dasjenige Systemmodell mit den besten Leistungseigenschaften wird für eine weitere Bearbeitung ausgewählt und mit SDL-Übersetzungswerkzeugen weitgehend automatisch in ein lauffähiges Objektsystem transformiert.
- Während der Laufzeit des Objektsystems kann mit einer spezifikationsgesteuerten Monitortechnik [2] die Leistung gemessen und mit den prognostizierten Daten des Systemmodells verglichen werden. Außerdem ermöglicht eine sog. Back-Annotation eine Verbesserung der Annahmen über die Werte von beeinflussenden Leistungs- und Echtzeitparametern in den SDL/MSC-Annotationen auf der Basis realistischer Meßdaten vorzunehmen.

3.2 Entwurf und Optimierung

Die Aktivitäten des Lehrstuhles bezüglich der Optimierung von Systemen umfassen die Paral-

leisierung von Kommunikationssoftware (in Zusammenarbeit mit der Siemens AG, Erlangen), die Parameteroptimierung von PROFIBUS-Installationen (in Zusammenarbeit mit der Siemens AG, Erlangen), die Optimierung von Kanban-Systemen (in Zusammenarbeit mit der IBM, Sindelfingen), die Optimierung der Steuerung von Fertigungsautomaten (in Zusammenarbeit mit der Siemens AG, München) sowie Untersuchungen zum Einsatz von genetischen Algorithmen zur heuristischen Systemoptimierung.

Mit Unterstützung der Siemens AG, Erlangen, wurde am Lehrstuhl die DSPL-Entwicklungsumgebung [5] erarbeitet, ein integrierter Ansatz zur Automatisierung des Entwurfs und der effizienten Realisierung dedizierter Anwendungen auf parallelen Systemen. Die DSPL-Entwicklungsumgebung unterstützt die automatische Abbildung der Anwendung auf ein gegebenes Multiprozessor-System, so daß die Leistung der generierten parallelen Implementierung ein Optimum erreicht.

Die Entwicklungsumgebung besteht aus einer grobgranularen Datenflußsprache und einer Reihe von integrierten Komponenten zur Ableitung von Modellen aus dem Datenflußprogramm, modellbasierten Optimierung des Entwurfs, insbesondere der Topologie des Zielsystems, der Zuordnung der Tasks auf die Prozessoren inklusive deren Bearbeitungsreihenfolge, der Granularität der Prozesse und der Platzierung von Puffern (bzw. deren Elimination), und der Umsetzung der Entwurfsentscheidungen mittels Transformationswerkzeugen.

Im Rahmen der modellbasierten Optimierung des Entwurfs wurden mehrere exakte und heuristische Optimierungsmethoden implementiert. Zur Bestimmung der exakten Lösung wurde ein Branch-and-Bound-Verfahren realisiert. Die implementierten Heuristiken umfassen Listenverfahren, Clustering-Verfahren und genetische Algorithmen. Echtzeitaspekte werden bei den derzeit implementierten Verfahren allerdings noch nicht berücksichtigt.

Zum Nachweis der Korrektheit der durchgeführten Transformationen wurde ein durchgängiger formaler Ansatz entwickelt.

3.3 Implementierung

Die DSPL-Entwicklungsumgebung ermöglicht die Generierung von optimiertem Code für Transputer-Netzwerke. Neben Anwendungen am Lehrstuhl, wurde die DSPL-Entwicklungsumgebung auch an der Universität Karlsruhe zur Realisierung eines verbindungslosen Netzwerkprotokolls (CLNP) auf der Parallel Protocol Engine, einem speichergekoppelten Transputer-Netzwerk, eingesetzt. Ein weiteres Anwendungsgebiet aus der Fertigungstechnik ist der Einsatz der Entwicklungsumgebung zur Optimierung einer verteilten Transputerbasierten Sensorsteuerung für einen Industrieroboter. Bei den meisten Realisierungen wurde ein T800-Transputersystem eingesetzt; im Rahmen des Teilprojekts B3 des Sonderforschungsbereiches 182 untersuchen wir seit Anfang 1995 auch die Anwendbarkeit der mit wesentlich schnelleren und flexibleren Kommunikations-Links ausgestatteten T9000-Transputer.

3.4 Messung und Bewertung

Zur hochgenauen Messung paralleler und verteilter Anwendungen wurde am Lehrstuhl ein universelles Hardwaremonitorsystem, der Zählmonitor 4 (ZM4) [6, 7] entwickelt. Der ZM4 unterstützt ereignisgesteuertes Monitoring von parallelen und räumlich verteilten Systemen. Zur Auswertung aufgezeichneter Meßspuren wurde die Auswertenumgebung SIMPLE entwickelt. Unterstützt werden die Analyse, Modellierung und Bewertung der gemessenen Systeme. ZM4 und SIMPLE werden an zahlreichen nationalen und internationalen Forschungseinrichtungen eingesetzt. Darunter sind u.a. das European Networking Center (ENC) der IBM in Heidelberg, die IBM in Rüschlikon, Schweiz, die Siemens AG, München, die Fudan Universität in Shanghai, VR China, die University of Maryland, USA und die Australian National University in Canberra, Australien.

3.5 Kommunikationsprotokolle

Der Lehrstuhl beschäftigt sich seit mehreren Jahren in Forschung und Lehre mit der formalen Spezifikation, Messung, Bewertung und Optimierung von Kommunikationsprotokollen. Neben den theoretischen Grundlagen, der Methodik und Werkzeugentwicklung für den Proto-

kollentwurf werden regelmäßig industriell eingesetzte Protokolle und Protokollhierarchien meßtechnisch untersucht und bewertet.

Beispiele dafür sind die Leistungsbewertung von echtzeitfähigen Feldbussystemen und deren Kommunikationsprotokollen, die prototypische Implementierung des MMS-Protokolles (Manufacturing Message Specification) auf einem Transputer-Netzwerk mit der DSPL-Entwicklungsumgebung und die Leistungsuntersuchung eines verteilten EURO-ISDN-Protokoll-Testsystems.

4 Methodik und Projektziele

Gegenstand des neuen Projektes ist die Entwicklung einer durchgängigen Methodik und eines Rapid-Prototyping-Systems zur Erstellung optimierter Echtzeitsysteme aus formalen Spezifikationen, sowie die Anwendung der Methodik auf die Realisierung einer Kommunikationsanschaltung für den Austausch von zeitkritischen Multimediadaten.

4.1 Eckpunkte

Die Eckpunkte des Projektes (vgl. hierzu Abbildung 1) sind:

- Formale Beschreibung sowohl von funktionalen Anforderungen als auch von Leistungsanforderungen an das zu realisierende System mittels annotierter SDL/MSK-Spezifikationen. Im Gegensatz zu den bisherigen Arbeiten werden Echtzeitkriterien festgelegt und durchgehend mitberücksichtigt.
- HW/SW-Systementwurf unter Berücksichtigung der Anforderungen an das System, inklusive harter Echtzeitbedingungen, durch modellbasierte Analyse- und Optimierungsverfahren.
- Abbildung der formalen Spezifikation auf Software bzw. Hardwarekomponenten eines zu erstellenden Rapid-Prototyping-Systems.
- Unterstützung der Produktentwicklung und Qualitätsnachweis durch spezifikationsbasiertes Monitoring und Bewertung der Prototypen. Realisierung einer Kommunikationsanschaltung für Echtzeitprotokolle unter Verwendung des erstellten Echtzeit-RP-System.

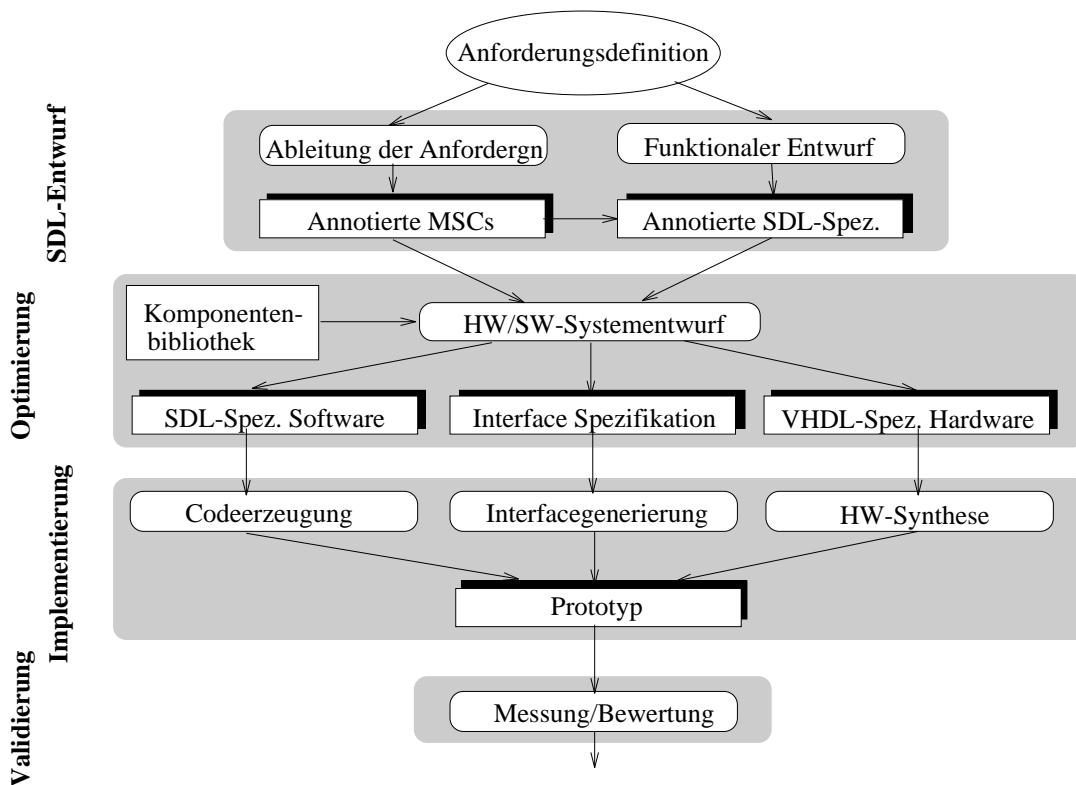


Abbildung 1: Vorgeschlagene Methodik der Systementwicklung

4.2 Methodik der Systementwicklung

Unsere Entwicklungsmethodik gliedert sich in vier Phasen (vgl. Abbildung 1): SDL-Entwurf, Optimierung, Implementierung und Validierung.

Ausgangsbasis für den *SDL-Entwurf* bildet die Anforderungsdefinition. Aus der Anforderungsdefinition wird die annotierte SDL-Spezifikation abgeleitet. Parallel dazu wird das dynamische Verhalten in Form von annotierten MSCs festgelegt mit Berücksichtigung von Echtzeit- und Kosten-Constraints. Das Ergebnis des SDL-Entwurfs ist sowohl eine vollständige SDL-Spezifikation als auch eine korrespondierende annotierte MSC-Hierarchie.

Die *Komponentenbibliothek* beschreibt die zur Verfügung stehenden Komponenten, wie Prozessoren, FPGAs, Speicherbausteine und Koppellemente: Neben der Beschreibung des funktionalen Verhaltens der Komponenten bzw. deren Fähigkeiten, SDL-Befehle und -Sequenzen zur realisieren, werden (gemessene) Leistungsdaten (z.B. Ausführungszeiten) sowie technische Randbedingungen (I/O-Ports, Kopplungsmöglichkeiten, etc.) gespeichert.

In der *Optimierungsphase* findet der HW/SW-Systementwurf statt. Zur Bestimmung eines möglichst guten Entwurfs kommen modellbasierte Analyse- und Optimierungsverfahren zur Anwendung. Bei der Analyse werden u.a. die Möglichkeiten der Realisierung des Systems unter Benutzung der in der Komponentenbibliothek zur Verfügung stehenden Komponenten untersucht. Im Anschluß erfolgt die Optimierung des Entwurfs entsprechend der Optimierungsziele und Randbedingungen. Nach der Bestimmung der Entwurfsentscheidungen erfolgt die Partitionierung der SDL-Spezifikation in die Teile, die in Hardware und Software realisiert werden sollen. Die SDL-Spezifikation der Software, die VHDL-Spezifikation der Hardware und die Spezifikation der HW/SW-Schnittstellen (Interfaces) sind das Ergebnis dieser Phase.

In der *Implementierungsphase* werden die in Hardware zu realisierenden Teile weitgehend automatisch in die Hardwarebeschreibungssprache VHDL umgesetzt und die entspre-

chenden Hardwarekomponenten erzeugt. Die in Software realisierten Teile werden aus SDL abgeleitet bzw. aus vorhandenen Bibliotheken zusammengesetzt. Die Schnittstellen werden gemäß ihrer Spezifikation erzeugt. Die Integration aller Komponenten ergibt den gewünschten Prototyp.

Bei der *Validierung* wird der erstellte Prototyp hinsichtlich seiner Funktionalität und seines Zeitverhaltens geprüft und die Einhaltung der Anforderungsdefinition validiert. Gleichzeitig erfolgt ein Vergleich, ggf. auch eine Aktualisierung der Komponentenbibliothek.

4.3 Anwendung: Echtzeitprotokolle

Zur Anpassung von multimedialen Kommunikationsanforderungen an ATM-Kommunikationsmechanismen müssen Transportdienste bereitgestellt werden, mit deren Hilfe Anwendungsprozesse ihre QoS-Anforderungen (Quality of Service) und Leistungscharakteristika für die Echtzeitkommunikation beschreiben und mit dem Diensterbringer aushandeln können. Dazu gehört u.a. die Charakterisierung der minimalen Zwischenankunftszeit einzelner Pakete, der maximalen Paketgröße, der maximalen Verzögerung und möglicher Abweichungen hiervon sowie des Pufferüberlaufverhaltens.

Neben anderen Vorschlägen, wie ST-II, SRP oder RSVP besitzt der Ansatz der ehemaligen Tenet-Gruppe am ICSI (International Computer Science Institut) in Berkeley den Vorteil, daß er explizit den Faktor Echtzeit in Form der obigen Parameter beschreiben und auch während des Kommunikationsvorganges überwachen lassen kann. Der Tenet-Protokollturn Scheme-1 besteht aus den Protokollen

- RCAP (Realtime Channel Administration Protocol) für Aufbau, Überwachung und Management von Echtzeitverbindungen mit garantiertem Antwortzeitverhalten auf Netz- und Transportebene,
- RMTP/RTIP (Realtime Message Transport/Internet Protocol) zur Übertragung von Echtzeitdaten unter Einhaltung der geforderten Leistungscharakteristika und
- CMTP (Continuous-Media Transport Protocol) zur Übertragung periodischer Rahmen,

evtl. variierender Größe wie bei komprimiertem Video.

Darüber hinaus existiert ein zweiter Protokollturn Scheme-2, der Multimedia-Gruppen-Kommunikation für wechselnde Sender-/Empfängerszenarien mit dynamischen Join/Leave-

Operationen unterstützt [1]. Innerhalb unseres Projekts dient die geschilderte Anwendung dazu, die von uns entwickelten Konzepte an einem zeitkritischen Multimedia-Echtzeitprotokoll zu erproben.

Literatur

1. A. Banerjea, D. Ferrari, B. Mah, M. Moran, D. Verma, H. Zhang. *The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences*. IEEE/ACM Transactions on Networking, 4(1), February 1996.
2. P. Dauphin, W. Dulz, F. Lemmen. Specification-Driven Performance Monitoring of DSL/MSC-specified Protocols. *Proc. 8th Intl. Workshop on Protocol Test Systems*, Evry, France. September 1995.
3. W. Dulz. A Framework for the Performance Evaluation of SDL/MSC-specified Systems. *Proc. Of European Simulation Multiconference*. 1996, Budapest, Hungary, June 1996.
4. A. Kalavade, E.A. Lee. A Global Critical/Local Phase Driven Algorithm for the Constrained Hardware/Software Partitioning Problem. *Proc. of Codes/CASHE '94, Third Intl. Workshop on Hardware/Software Codesign*, Grenoble, France, Sept. 1994.
5. A. Mitschele-Thiel. *Die DSPL-Entwicklungsumgebung — Ein automatischer Ansatz zur Abbildung und effizienten Realisierung dedizierter Anwendungen auf parallele Systeme*. Fortschritt-Berichte VDI, Reihe 10: Informatik/ Kommunikationstechnik, Nr. 315, VDI Verlag, 1994.
6. R. Hofmann. The Distributed Hardware Monitor ZM4 and Its Interface to MEMSY. In A. Bode and M. Dal Cin, editors, *Parallel Computer Architectures: Theory, Hardware, Software, Applications*, pages 66-79. Springer Lecture LNCS 732, Berlin et al., March 1993.
7. R. Hofmann and R. Klar. ZM4 — a Universal Distributed Hardware Monitor. In *Proceedings of APPT'95, Int. Workshop on Advanced Parallel Processing Technologies*, pages 64-71, Beijing, China, September 26-27, 1995. Northern Jiaotong University, Publishing House of Electronics Industry.
8. Project INSYDE. *Application Guidelines*. Technical report, CEC Deliverable Number P8641/HUB/INF/DS/P/D1.2/b1 1995.
9. Telelogic Malmö AB: *SDT 3.0 User's Guide, SDT 3.0 Reference Manual*, 1995.
10. Project TOPIC. *State of the Art and Design of Verification Environment*. Technical report, CEC Deliverable Number R2088/TUB/PRZ/DS/P/002/b2, 1992.
11. Verilog. GEODE - Technical Presentation, 1994.
12. K. Waldschmidt, A. Bode, U. Brüning, B.M. Chapman, M. Dal Cin, W. Händler, F. Hertweck, U. Herzog, F. Hofmann, R. Klar, C.-U. Linster, W. Rosenstiel, H.-J. Schneider, J. Wedeck, and H. Zima. *Parallelrechner: Architekturen, Systeme, Werkzeuge*. Leitfäden der Informatik. Teubner Verlag, Stuttgart, 1995.