

# SPI-Workbench – Modellierung, Analyse und Optimierung eingebetteter Systeme

Ch. Haubelt, M. Jersak, K. Richter, K. Strehl, D. Ziegenbein  
R. Ernst, J. Teich und L. Thiele  
haubelt@cs.fau.de

**Abstract:** Das SPI-Modell dient zur Repräsentation von heterogenen eingebetteten Systemen. Basierend auf diesem Modell wurde in einem Forschungsverbund zwischen der TU Braunschweig, der Universität Erlangen-Nürnberg und der ETH Zürich die systemweite Analyse und Optimierung eingebetteter Systeme untersucht und in der SPI-Workbench integriert. Besonderes Augenmerk wurde auf die domänenübergreifende Integration von Systemteilen gelegt. Dieser Beitrag wird die Ergebnisse dieses erfolgreichen Projektes zusammenfassen: Zunächst wird die SPI-Workbench sowie die Idee von SPI eingeführt. Anschließend wird die Analyse des Zeitverhaltens diskutiert. Zum Abschluss werden neue Ansätze zur Entwurfsraumexploration basierend auf dem SPI-Modell vorgestellt. Die in dem SPI-Projekt entstandenen Ergebnisse stellen über den Projektzeitraum hinaus eine wesentliche Grundlage für State-of-the-Art Forschung im Bereich Entwurf heterogener eingebetteter Systeme dar.

## 1 Einleitung

Komplexe eingebettete Systeme wie mobile Kommunikationsgeräte, Industriesteuerungen, Medizintechnik, etc. sind von Natur aus meist *heterogen*. Ein Aspekt der Heterogenität liegt in den unterschiedlichen Berechnungsmodellen, die diese Systeme oder Teile hieraus beschreiben. So beschreibt ein Modell eines Audiosystems überwiegend die Transformation von Eingangsströmen in Ausgangsströme, während eine Industriesteuerung auf Eingangsereignisse reagiert. Ein Beispiel für ein System, das sowohl reaktive als auch transformative Komponenten enthält, ist das Mobiltelefon.

Im Zusammenhang mit gemischt reaktiven/transformativen Systemen liegen Probleme in den Bereichen der *Analyse*, *Synthese*, *Optimierung* und *Verifikation*. Die hier vorgestellten Ergebnisse sind in einer Kooperation der TU Braunschweig (Prof. Ernst), der Universität Erlangen-Nürnberg (Prof. Teich) und der ETH Zürich (Prof. Thiele) entstanden. Innerhalb dieser Kooperation wurden Modelle und Methoden zum Entwurf gemischt reaktiv/transformativer eingebetteter Systeme entwickelt. Das Ergebnis ist das *SPI-Modell* (System Property Intervals), das in der Lage ist, von vielen bekannten Modellen zu abstrahieren. Für dieses Modell wurden Methoden entwickelt, die entweder direkt aus bekannten Methoden abgeleitet oder gänzlich neu entwickelt wurden. Die Philosophie besteht also darin, sowohl domänenspezifische Modelle und Analyseverfahren zu integrieren als auch neue Modelle einfach in das resultierende Entwurfssystem einbringen zu können (siehe

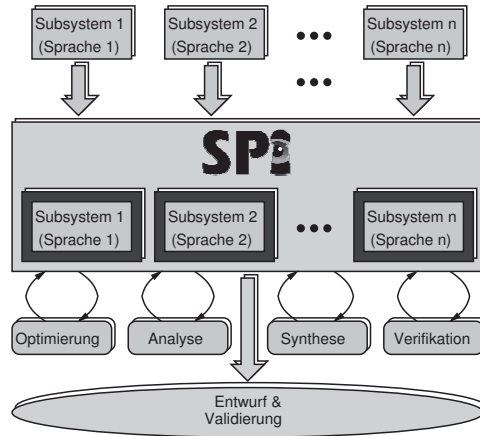


Abbildung 1: Domänenspezifischen Spezifikationen werden in dem SPI-Modell gekapselt. Anschließend kann eine globale Analyse, Synthese, Optimierung oder Verifikation des Modells erfolgen.

Abbildung 1). Die Ergebnisse der durchgeführten theoretischen Untersuchungen wurden implementiert und prototypisch in der SPI-Workbench integriert.

## 2 Das SPI-Modell

Zu Beginn des SPI-Projektes wurde das SPI-Modell (System Property Intervals) definiert [ZRE<sup>+</sup>02, STG<sup>+</sup>01, Str00, Zie02]. Die Besonderheit des SPI-Modells liegt vor allem in den Bereichen a) Modellierung von Kontroll- und Datenfluss und b) Modellierung von unsicherem Verhalten durch Intervalle.

Ein einfaches SPI-Modell, bestehend aus drei Prozessen ( $P_1$ ,  $P_2$  und  $P_3$ ) welche über zwei Kanäle ( $C_1$  und  $C_2$ ) miteinander kommunizieren, ist in Abbildung 2 gegeben. Sowohl den Kanälen  $C_i$  als auch den Prozessen  $P_i$  sind Latenzzeiten  $lat_{C_i}$  bzw.  $lat_{P_i}$  zugeordnet, welche die Bearbeitungszeit eines Kanals bzw. Prozesses darstellen. Weiterhin sind Parameter  $s_{C_i}$  und  $r_{C_i}$  den Kanälen  $C_i$  zugeordnet. Diese geben die Anzahl der produzierten bzw. konsumierten Daten bei Ausführung des jeweiligen Prozesses an. Die Anzahl der Marken, die in einem Kanal  $C_i$  gespeichert sind, werden durch den Parameter  $d_{C_i}$  modelliert.

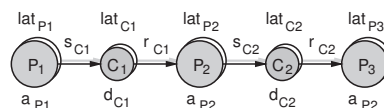


Abbildung 2: Einfaches Beispiel eines SPI-Prozessgraphen. Die Prozesse  $P_1$ ,  $P_2$ , und  $P_3$  kommunizieren über Kanäle  $C_1$ ,  $C_2$ . Die Parameter  $lat$ ,  $s$ ,  $r$ , und  $d$  sind durch Intervalle spezifiziert. Die Aktivierungsfunktionen  $a_{P_i}$  beeinflussen die Ausführung eines Prozesses  $P_i$ .

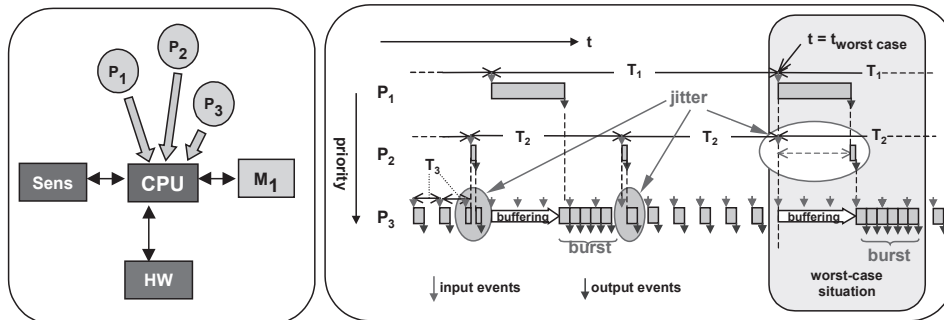


Abbildung 3: Obwohl das Betriebssystem die Prozesse  $P_1$ ,  $P_2$  und  $P_3$  strikt periodisch mit den Perioden  $T_1$ ,  $T_2$  und  $T_3$  aktiviert, führt die Ausführungssequenz zu Bursts an den Ausgängen.

Die Aktivierung eines Prozesses  $P_i$  wird durch die Aktivierungsfunktion  $a_{P_i}$  bestimmt. Sämtliche Parameter  $lat$ ,  $s$ ,  $r$  und  $d$  sind durch Intervalle repräsentiert. Dies bedeutet, dass entweder die genauen Werte nicht bekannt sind oder die Parameter ungenau spezifiziert wurden. Das SPI-Modell bildet die Basis für die oben genannten Kooperationspartner.

### 3 Zeitanalyse

Die Heterogenität eingebetteter Systeme ist nicht nur auf deren Verhaltensmodellierung beschränkt, sondern findet sich in zunehmenden Maße auch bei den zu Grunde liegenden Architekturen wieder. Hier kann ein Trend in Richtung verteilter Multiprozessorarchitekturen festgestellt werden, welche oftmals viele programmierbare Prozessoren, Speicherbausteinen und IP-Komponenten (Intellectual Property) auf einem einzelnen Chip integrieren. Bei diesen Architekturen wird das Ressourcen-sharing oftmals durch Einsatz von Echtzeitbetriebssystemen realisiert.

Die Analyse des zeitlichen Verhaltens eines eingebetteten Systems, welches eine verteilte Multiprozessorrealisierung als Zielarchitektur verwendet, ist eine der heutigen Herausforderungen im Entwurf eingebetteter Systeme. Wie in Abbildung 3 dargestellt, kann der Prozess  $P_1$  mehrere aufeinander folgende Ausführungen von Prozess  $P_3$  verzögern. Erst anschließend wird Prozess  $P_3$  burstartig seine Eingaben abarbeiten. Selbst bei diesem kleinen Beispiel erkennt man die Probleme bei der Analyse des Zeitverhaltens eingebetteter Systeme. State-of-the-Art in der Zeitverifikation ist die Verwendung simulationsbasierter Methoden. Allerdings haben simulationsbasierte Analyseansätze den Nachteil, dass diese nicht in der Lage sind, das kritische Zeitverhalten eines Systems zu bestimmen.

Im Rahmen des SPI-Projektes wurde ein anderer Ansatz mit dem Namen SymTA/S entwickelt und untersucht [RJE03, Jer05, Ric05], welcher die Analyse einer großen Teilmenge von SPI-Modellen erlaubt: Die grundlegende Idee basiert auf der Kopplung unterschiedlicher Ablaufplanungsanalyseverfahren durch sog. *Ereignisströme*. Ereignisströme beschreiben das mögliche Eingabe/Ausgabe-Verhalten von Prozessen und werden durch

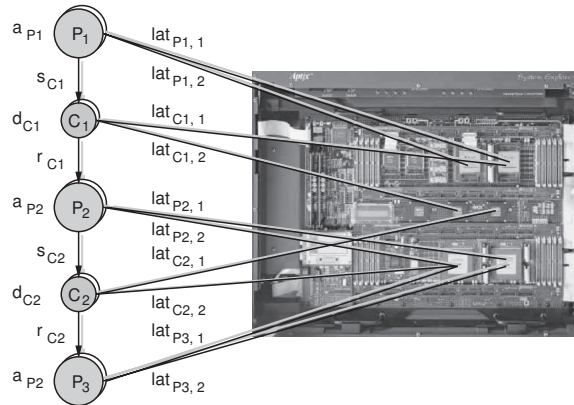


Abbildung 4: Beispiel einer ressourcenbeschränkten Spezifikation. Die Prozesse (Kanäle) des SPI-Prozessgraphen können auf Komponenten der Architektur abgebildet werden. Die Ausführungszeiten  $lat$  hängen von der Bindung der Prozesse auf die Komponenten ab.

geeignete Ereignismodelle, wie z.B. *periodisch* oder *sporadisch*, charakterisiert. SymTA/S unterstützt hierbei die Kombination und Integration von unterschiedlichen Analyseansätzen. Bei dieser Kombination kann es aber zu Inkompatibilitäten bezüglich der verwendeten Ereignisströme kommen. Dieses Problem löst SymTA/S durch die Verwendung von sog. *EMIFs* (Event Model InterFaces) und sog. *EAFs* (Event Adaptation Functions).

Ein weiterer Ansatz zur Zeitanalyse in eingebetteten Echtzeitsystemen wurde basierend auf dem sog. *Network Calculus* entwickelt und untersucht [TCGK02]. Hierbei werden Prozesse durch sog. *Arrival Curves* und Ressourcen durch sog. *Service Curves* charakterisiert. Die Zeitanalyse erfolgt schließlich über ein sog. *Schedulingnetwork*.

## 4 Entwurfsraumexploration

Das Problem der Hardware/Software-Partitionierung beschreibt die Aufgabe, Prozesse an Ressourcen zu binden (Abbildung 4). Basierend auf dem SPI-Modell mit hierarchischer Verfeinerungen wurden zwei Ansätze untersucht: a) die *hierarchische Partitionierung* mit evolutionären Algorithmen und b) die sog. *Pareto-Front-Arithmetik* [Hau05]. Die hierarchische Partitionierung codiert das zu partitionierende Problem als ganzes und verwendet evolutionäre Algorithmen zur Suche optimaler Lösungen. Die genetischen Operatoren beeinflussen hierbei auch die Auswahl der Verfeinerungen für Prozesse.

Bei der Pareto-Front-Arithmetik werden zunächst die Verfeinerungen der Prozesse einzeln optimiert und später diese Lösungen kombiniert. Da diese Kombination im Zielfunktionsraum vorgenommen wird, bezeichnen wir diese Verfahren als *Pareto-Front-Arithmetik*. Das Ergebnis sind approximierte Lösungen des Gesamtproblems. Dieses Verfahren führt im Allgemeinen jedoch weder zu gültigen noch zu optimalen Ergebnissen. Dennoch sind diese Ergebnisse hilfreich, um das oben genannte evolutionäre Verfahren zu initialisieren.

Im Rahmen des SPI-Projektes wurden die Verfahren zur Optimierung bei der Hardware/Software-Partitionierung eingebetteter Systeme um Verfahren zur sog. *stochastischen Pareto-Front-Analyse* erweitert [Tei01]. Hierbei wird nicht mehr mit exakten Werten sondern mit Intervallen und Wahrscheinlichkeitsdichtefunktionen das Optimierungsproblem gelöst. Die Idee hierbei ist, dass Zielgrößen im Allgemeinen nicht exakt bekannt sind sondern, wie im SPI-Modell, meist nur untere und obere Schranken. Diese Verfahren wurden erfolgreich in die Pareto-Front-Arithmetik integriert [Hau05].

## 5 Zusammenfassung

In diesem Bericht wurden einige zentrale Ergebnisse des SPI-Projekt zum Entwurf heterogener eingebetteter Systeme skizziert. Das SPI-Projekt war ein Verbundforschungsprojekt der TU Braunschweig, der Universität Erlangen-Nürnberg und der ETH Zürich und wurde im Rahmen des DFG-Schwerpunktprogramms „Entwurfsmethoden für Eingebettete Systeme“ durch die DFG gefördert.

## Literatur

- [Hau05] Christian Haubelt. *Model-Based Design Space Exploration for Embedded Systems – A System Level Approach*. Dissertation, Universität Erlangen-Nürnberg, 2005. In Druck.
- [Jer05] Marek Jersák. *Compositional Scheduling Analysis Using Standard Event Models*. Dissertation, Technische Universität Braunschweig, 2005.
- [Ric05] Kai Richter. *Compositional Performance Analysis for Complex Embedded Applications*. Dissertation, Technische Universität Braunschweig, 2005. In Druck.
- [RJE03] Kai Richter, Marek Jersak und Rolf Ernst. A Formal Approach to MpSoC Performance Verification. *IEEE Computer*, 36:60–67, April 2003.
- [STG<sup>+</sup>01] Karsten Strehl, Lothar Thiele, Matthias Gries, Dirk Ziegenbein, Rolf Ernst und Jürgen Teich. FunState - An Internal Design Representation for Codesign. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(4):524–544, August 2001.
- [Str00] Karsten Strehl. *Symbolic Methods Applied to Formal Verification and Synthesis in Embedded Systems Design*. Dissertation, ETH Zürich, Februar 2000.
- [TCGK02] Lothar Thiele, Samarjit Chakraborty, Matthias Gries und Simon Künzli. Design Space Exploration of Network Processor Architectures. *Network Processor Design: Issues and Practices*, 1:55–89, Oktober 2002.
- [Tei01] Jürgen Teich. Pareto-Front Exploration with Uncertain Objectives. In *Proceedings of the First Int. Conf. on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science (LNCS)*, Jgg. 1993, Seiten 314–328, Zurich, Switzerland, Marz 2001.
- [Zie02] Dirk Ziegenbein. *A Compositional Approach to Embedded System Design*. Dissertation, Technische Universität Braunschweig, Marz 2002.
- [ZRE<sup>+</sup>02] Dirk Ziegenbein, Kai Richter, Rolf Ernst, Lothar Thiele und Jürgen Teich. SPI - A System Model for Heterogeneously Specified Embedded Systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 10(4):379–389, August 2002.