

OOCOSIM - Eine objekt-orientierte Co-Designmethode für eingebettete Hardware/Softwaresysteme

*Frank Oppenheimer, ⁺Michael Kersten, ⁺Wolfgang Nebel

*Kuratorium OFFIS e.V.
Eingebettete Hardware/Software-Systeme
frank.oppenheimer@offis.de

⁺Carl v. Ossietzky Universität Oldenburg
Fakultät II – Department für Informatik
Eingebettete Hardware/Software-Systeme
{kersten|nebel}@informatik.uni-oldenburg.de

Abstract: Die steigende Komplexität von eingebetteten Systemen und die daraus resultierenden Designprobleme unterstreichen die Notwendigkeit von verbesserten Entwurfsmethoden. Dieser Artikel präsentiert eine objekt-orientierte Entwurfsmethode namens OOCOSIM, die diese Probleme adressiert. Für die Analyse bietet sie die graphische Notation namens HRT-HOOD+. Diese Beschreibungsform ermöglicht den Überblick und die Analyse von komplexen HW/SW-Systemen. Darüber hinaus wurde untersucht, inwieweit UML geeignet ist, die Beschränkungen von HRT-HOOD+ zu überwinden. Zur Überprüfung des Verhaltens der Systemmodelle wurde ein Konzept zur zeitsynchronen Co-Simulation entwickelt.

1 Einführung

Hauptziel des DFG-geförderten Projekts "OOCOSIM" war die Konzeption und Implementierung einer objektorientierten Entwurfsmethode für eingebettete Realzeitsysteme, die von der abstrakten Spezifikation bis zur Implementierung des zu entwickelnden Systems reicht. Von besonderer Bedeutung war dabei die Möglichkeit der abstrakten Spezifikation und Verfeinerung, um auch komplexe Systeme beherrschbar zu machen. Zur Vermeidung häufig auftretender Designfehler an der Hardware/Software-Schnittstelle, wurde eine flexible, objektbasierte Entwurfsmethode für diesen Aspekt entwickelt. Die formalisierte Beschreibung der Schnittstelle in einer eigens hierfür entwickelten Sprache namens ComiX (Abschnitt 2.4) erlaubt die Überprüfung zahlreicher Konsistenzeigenschaften. Außerdem ermöglicht die Beschreibung eine automatische Generierung der Hardware- und Softwareimplementierung der Schnittstelle. Um Aussagen über das Echtzeitverhalten von komplexen Systemen treffen zu können, wurde ein Konzept zur zeitsynchronen HW/SW-Co-Simulation (Abschnitt 2.5) in OOCOSIM integriert.

2 OOCOSIM im Überblick

Die OOCOSIM-Methode beginnt mit der abstrakten Modellierung auf Systemebene. Dabei geht es letztlich darum, durch schrittweise Verfeinerung eine Dekomposition des Systems in Software-Komponenten, Hardware-Komponenten sowie Kommunikations-Komponenten zu erreichen. Dazu wurden für OOCOSIM, basierend auf existierenden Standards, zwei alternative grafische Modellierungssprachen definiert; HRT-HOOD+ (Abschnitt 2.2) und UML-RTES (Abschnitt 2.3). Diese Sprachen dienen als Ausgangspunkt für den OOCOSIM Designflow (Abbildung 1). Transformationen von der Systemspezifikation zum Validierungsmodell und zum Implementierungsmodell garantieren dabei die Durchgängigkeit.

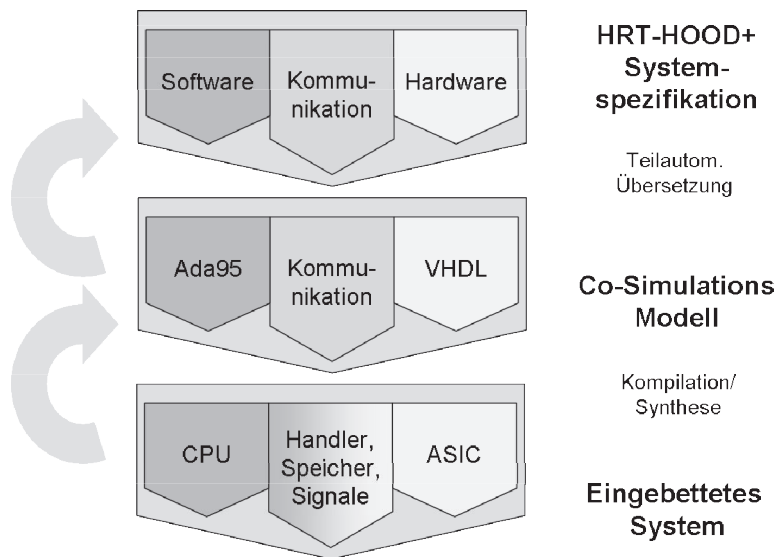


Abbildung 1: OOCOSIM Designflow

Die Codegenerierung basiert auf einer in diesem Kontext entwickelten Template-Sprache namens TempliX [OZN01], auf die in diesem Artikel aus Platzgründen leider nicht näher eingegangen werden kann. Durch Kompilation des Ada95-Codes und Synthese des VHDL-Codes wird schließlich die Implementierung des eingebetteten Systems erzeugt.

2.1 Spezifikation auf abstrakten Ebene

Für die abstrakte Modellierung wurden zwei Ansätze näher untersucht: zum einen die HRT-HOOD+ Methode und zum anderen der UML-RTES Ansatz. HRT-HOOD+ stellt eine Erweiterung – insbesondere um die im Weiteren vorgestellten Schnittstellenobjekte – der HRT-HOOD Methode [BW95] dar. UML-RTES erweitert UML [Om01] um HRT-HOOD+-ähnliche Konzepte, profitiert aber zusätzlich von dem weitergehenden objekt-orientierten Modell der UML und der Möglichkeit zur Verhaltensspezifikation.

2.2 Der HRT-HOOD+ Ansatz

Neu an der HRT-HOOD+ Methode sind System-Objekte zur Beschreibung von Teilsystemen, Hardware-Objekte zur Repräsentation von Hardwarekomponenten und Kommunikations-Objekte zur Modellierung der HW/SW-Schnittstelle von eingebetteten Systemen auf abstrakter Ebene. Letztere stehen in folgenden Varianten zur Verfügung:

- *Memory-Objects* dienen der Modellierung von Datenflüssen (shared-memory),
- *Asynchronous-Signals* beschreiben asynchrone Hardware-Events (interrupts),
- *Asynchronous-Objects* verbinden beide Konzepte miteinander.

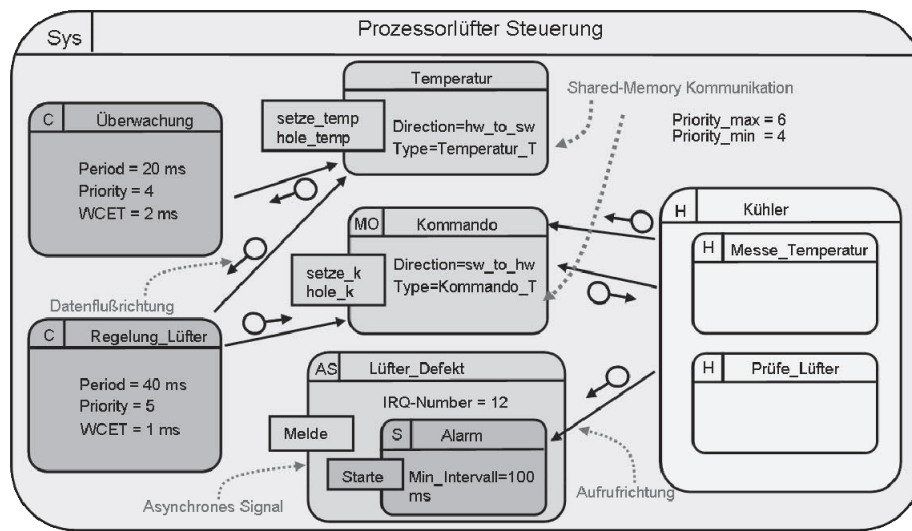


Abbildung 2: Hardware/Software-Interface in HRT-HOOD+

Abbildung 2 veranschaulicht am Beispiel einer einfachen Lüfterregelung die Modellierung in HRT-HOOD+. Das Subsystem wird durch das *System_Object* „Prozessorlüfter Steuerung“ gegen die Umgebung abgegrenzt. Darin enthalten sind:

- Zwei zyklische Software-Objekte (*Überwachung*, *Regelung_Lüfter*),
- Drei Hardware-Objekte (*Kühler*, *Messe_Temperatur*, *Prüfe_Lüfter*),
- Drei Kommunikations-Objekte (*Temperatur*, *Kommando*, *Lüfter_Defekt*).

Bereits im HRT-HOOD+ Modell werden wesentliche Aspekte der Hardware/Software-Schnittstelle modelliert. Im obigen Beispiel erlaubt das *Memory-Object* *Kommando* die shared-memory Kommunikation zwischen der HW-Komponente *Kühler* und der SW-Komponente *Regelung_Lüfter*. Die Methodenschnittstelle ermöglicht den gekapselten Zugriff auf Zustände des Objekts. Das *Asynchronous-Signal* *Lüfter_Defekt* modelliert die unmittelbare Benachrichtigung der Software, falls die Hardware (*Prüfe_Lüfter*) einen defekten Lüfter erkennt. Für eine ausführlichere Beschreibung sei auf [Op05] verwiesen.

2.3 Der UML-RTES Ansatz

Der UML-RTES Ansatz ist eine Erweiterung einer Teilmenge der populären Unified Modeling Language (UML) um spezielle Aspekte zur HW/SW-Schnittstellenmodellierung, sowie zur Modellierung von Realzeiteigenschaften. Konkret erfordert dies die Definition eines UML Profils für Klassen-, Zustands- und Kollaborationsdiagramme sowie HRT-HOOD+ nachempfundene Stereotypen und tagged values. Die Möglichkeit der Realzeitmodellierung wurde durch die Erweiterung des UML Metamodells um Zeit-Attribute erreicht. Durch diese Erweiterungen erlaubt UML-RTES die Formulierung komplexer Zeitbedingungen in OCL ([Ke04]).

2.4 Verfeinerung der Kommunikation durch ComiX

Nach der Systemspezifikation wird nun die Kommunikation auf Basis der im Projekt entwickelte Sprache ComiX (Communication interfaces in XML) verfeinert. Abbildung 3 zeigt eine ComiX Beschreibung, insbesondere die Repräsentation des *Memory-Objects Kommando* sowie des Speichermappings.

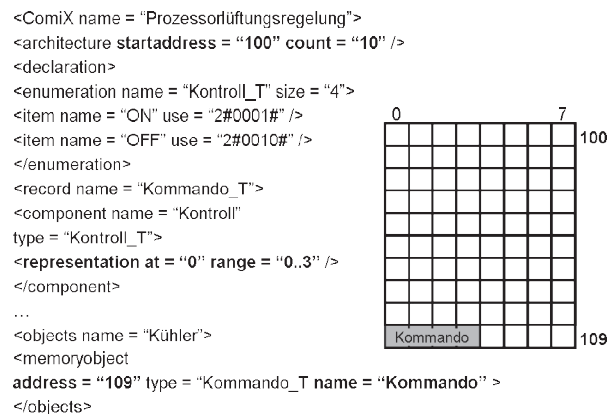


Abbildung 3: Schnittstellenbeschreibung in ComiX.

Durch einen Codegenerator wird die ComiX-Beschreibung in Softwareanteile (Ada95), Hardwareanteile (VHDL) und eine Dokumentation (LaTeX) übersetzt.

2.5 Hardware/Software-Co-Simulation

Das grundlegende Ziel in OOCOSIM war es, eine zeitsynchrone Co-Simulation der SW- und HW-Komponenten zu ermöglichen. Dazu wird eine gemeinsame Uhr eingeführt und in Software- und Hardwaremodell eine gemeinsame Ereignisverwaltung (Abbildung 4) verwendet. Als Konsequenz ergibt sich, dass auf beiden Seiten der HW/SW-Co-Simulation zur Synchronisation wechselseitig gewartet werden muss. Eine Herausforderung stellt dabei der Umgang mit asynchronen Hardwareereignissen dar, da diese zu beliebigen Zeitpunkten auftreten können und unmittelbar behandelt werden müssen.

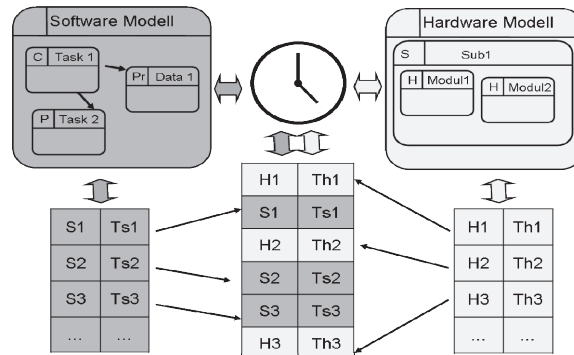


Abbildung 4: Gemeinsame Ereignisverwaltung in der OOCOSIM Simulation

Weitere Aspekte der Co-Simulation wurden in [OS99] bzw. [Op05] veröffentlicht.

3 Zusammenfassung

Den Rahmen des OOCOSIM Projektes bildete die Untersuchung einer Gesamtmethode zum Entwurf eingebetteter Systeme, basierend auf dem objektorientierten Modellierungsparadigma. Diese Methode bietet die Möglichkeit zur abstrakten Spezifikation auf Systemebene, wozu HRT-HOOD bzw. UML erweitert wurden. Besonderes Augenmerk liegt dabei auf den Realzeiteigenschaften und der Nutzung der Objektorientierung für die Modellierung von Hardware-/Software-Schnittstellen. Dazu wurden Kommunikations-Objekte und die Sprache ComiX eingeführt. Zur Validierung von Systemmodellen wurde eine Umgebung zur zeitsynchronen HW/SW-Co-Simulation erstellt. Um eine Durchgängigkeit der Methodik zu erreichen, wurden Konzepte zur Codegenerierung aus ComiX entwickelt und prototypisch implementiert.

4 Literaturverzeichnis

- [BW95] Burns, A. und A. Wellings: HRT-HOOD A Structured Design Method for Hard Real-Time Systems. Elsevier, 1995.
- [Ru95] Rumbaugh, J.: OMT: The Object Model. Journal of Object Oriented Programming, 1995.
- [OSN99] Oppenheimer, F.; Schumacher G.; Nebel, W.: OOCOSIM - Objektorientierte Spezifikation und Simulation eingebetteter Realzeitsysteme: it+ti., 41(1999), Nr. 2, 1999.
- [Om01] OMG. OMG Unified Modeling Language Specification (version 1.4). Technical report, OMG, <http://www.omg.org>, September 2001.
- [OZN01] Oppenheimer, F.; Zhang, D.; Nebel, W.: Modelling Communication Interfaces with ComiX, Proceedings of the Ada-Europe 2001. - Berlin u.a.: Springer, 2001.
- [Ke04] Kersten M. et. al.: Erweiterung der UML um Zeitannotationen zur Analyse des Zeitverhaltens reaktiver Systeme, In: Drechsler, R.: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen. - Aachen: Shaker, 2004.
- [Op05] Oppenheimer, F.: OOCOSIM - An Object-oriented Co-Design Method for Embedded HW/SW Systems, Dissertation, Universität Oldenburg, 2005.