



Doctoral Thesis

Compositional design and analysis of distributed, cyclic, and adaptive embedded real-time systems

Author(s):

Stoimenov, Nikolay Nikolaev

Publication Date:

2011

Permanent Link:

<https://doi.org/10.3929/ethz-a-006543550> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

DISS. ETH NO. 19619

Compositional Design and Analysis of Distributed, Cyclic, and Adaptive Embedded Real-Time Systems

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

NIKOLAY NIKOLAEV STOIMENOV

Bachelor of Computer Science, First Class Honours,
The University of Adelaide, Australia

born April 25, 1979

citizen of Bulgaria

accepted on the recommendation of

Prof. Dr. Lothar Thiele, examiner
Prof. Dr. Petru Eles, co-examiner

2011

Abstract

Embedded systems are computer systems that are deeply integrated in and interact with the physical world. The physical world often imposes strict timing constraints on these systems. Therefore, the correct operation of such systems depends not only on the values of the produced results, but also on their timing. Such systems are called real-time systems.

Methods for system-level performance analysis play an integral part during the early design phase of embedded real-time systems. They support the analysis of various non-functional performance characteristics, and alleviate the choice of important design decisions before much time and resources are invested in detailed implementations.

Compositional formal methods for performance analysis are able to quickly provide hard upper and lower bounds on performance characteristics. However, such analytical methods are limited in their scope and accuracy as they cannot incorporate many system details in the analysis.

Interface-based design has been proposed in order to unify the steps of designing a system and analyzing it. It supports the paradigm for correct-by-construction design in the domain of embedded real-time systems. It can significantly shorten the design time of complex distributed embedded real-time systems.

Recently, a method for Modular Performance Analysis based on Real-Time Calculus has been proposed that is also connected with the principles of interface-based design in the framework of Real-Time Interfaces. Both of them support the analysis and design of complex distributed embedded real-time systems. This thesis builds on these results and extends them in several directions. The main contributions of this work are summarized as follows:

- A novel framework for interface-based design of distributed embedded real-time systems is proposed. It includes properties

such as incremental design, independent-implementability, and refinement. It unifies many existing compositional performance analysis and interface-based design methods.

- Novel models and methods for interface-based design are proposed that support the analysis and design of distributed embedded real-time systems which have buffer overflow and underflow constraints, end-to-end delay constraints, variable execution demands of tasks, and complex resource sharing policies.
- A novel method for compositional performance analysis of marked graphs is proposed that can be used for distributed embedded real-time systems with cyclic data dependencies, finite buffers with blocking write semantics, variable execution demands of tasks, non-deterministic event streams and resource behaviors, and complex resource sharing policies.
- A novel method for mode change performance analysis of multi-mode embedded real-time systems is proposed that can be used for systems with non-deterministic event streams and resource behaviors, variable execution demands of tasks, complex resource sharing policies, and various mode change protocols.
- A novel scheduling server based on time division multiple access is proposed that can be reconfigured during run-time, and can guarantee the real-time properties of applications during reconfigurations.

Zusammenfassung

Eingebettete Systeme sind Computersysteme, die in eine physikalische Umgebung eingebunden sind und mit dieser intensiv interagieren. Oft müssen eingebettete Systeme strenge Zeitbedingungen einhalten, die von der physikalischen Umgebung auferlegt werden. Die korrekte Ausführung des Systems hängt in solchen Fällen nicht nur von den berechneten Ergebnissen ab, aber auch vom Zeitpunkt, an dem die Ergebnisse produziert werden. Solche Systeme werden allgemein als Echtzeit-Systeme bezeichnet.

In frühen Entwicklungsphasen von eingebetteten Echtzeit-Systemen spielen Methoden zur Leistungsbewertung auf Systemebene eine wesentliche Rolle. Sie ermöglichen die Analyse von verschiedenen Leistungskriterien wie beispielsweise die Ausführungszeit des Systems und erleichtern das Treffen von wichtigen Entwurfsentscheidungen bevor Zeit und Ressourcen für die eigentliche Implementierung des Systems aufgewendet werden.

Modulare formale Methoden zur Leistungsbewertung können in kurzer Zeit sichere obere und untere Schranken für Leistungskriterien von Systemen bestimmen. Diese analytischen Verfahren sind allerdings in ihrem Anwendungsbereich und ihrer Genauigkeit eingeschränkt, da sie viele Systemdetails nicht in die Analyse einbinden können.

Die interface-basierte Entwicklung von eingebetteten Echtzeit-Systemen wurde eingeführt, um die beiden Schritte des Entwurfs und der Analyse eines Systems zu vereinen. Diese Art von Entwicklung folgt dem Paradigma des correct-by-construction, d.h. Systeme werden so entworfen, dass die Leistungskriterien garantiert erfüllt sind. Interface-basierte Entwicklung kann zudem die Entwicklungszeit von komplexen verteilten Echtzeit-Systemen massgeblich verkürzen.

Vor kurzem wurde die Modular Performance Analysis, eine auf Real-Time Calculus basierte Methode zur Leistungsbewertung vorgestellt. Diese Methode ist eng mit dem Verfahren der Real-Time Interfaces

verknüpft, die die Prinzipien der interface-basierten Entwicklung umsetzt. Beide Verfahren unterstützen die Analyse und den Entwurf von komplexen verteilten eingebetteten Echtzeit-Systemen. Diese Dissertation baut auf beide Verfahren auf und erweitert sie in mehrere Richtungen. Die wichtigsten Beiträge dieser Arbeit können wie folgt zusammengefasst werden:

- Ein neues Framework zur interface-basierten Entwicklung von verteilten eingebetteten Echtzeit-Systemen wird vorgestellt. Das Verfahren erfüllt die Eigenschaften des inkrementellen Entwurfs, der unabhängigen Implementierbarkeit und der Verfeinerung. Es vereint verschiedene existierende modulare Methoden zur Leistungsbewertung und interface-basierten Entwicklung.
- Neue Modelle und Methoden für interface-basierte Entwicklung werden vorgeschlagen, die die Analyse und den Entwurf von verteilten eingebetteten Echtzeit-Systemen mit folgenden Eigenschaften ermöglichen: Einschränkungen bzgl. Puffer-Überlauf/Unterlauf, Einschränkungen bzgl. End-to-end Latenzen, variable Ausführungszeiten von Tasks, Ressourcen-Sharing anhand komplexer Verfahren.
- Eine neue Methode zur modularen Analyse von markierten Graphen wird vorgestellt, die für verteilte eingebettete Echtzeit-Systeme mit einer oder mehreren der folgenden Eigenschaften eingesetzt werden kann: Zyklische Abhängigkeiten zwischen Komponenten, endliche Puffer mit blocking-write Semantik, variable Ausführungszeiten von Tasks, nicht-deterministische Ereignisströme oder Ressourcenverfügbarkeit, Ressourcen-Sharing anhand komplexer Verfahren.
- Ein neues Verfahren zur Analyse von Mode-Wechseln in eingebetteten Echtzeitsystemen mit mehreren Ausführungsmodi wird eingeführt. Das Verfahren unterstützt nicht-deterministische Ereignisströme und Ressourcenverfügbarkeit, variable Ausführungszeiten von Tasks, komplexe Verfahren für Ressourcen-Sharing sowie verschiedene Protokolle für Mode-Wechsel.
- Ein neuer auf Zeitmultiplexing aufbauender Scheduling-Server wird vorgestellt, der zur Laufzeit umkonfiguriert werden kann. Der Server garantiert die Einhaltung der Echtzeit-Anforderungen von Applikationen auch während der Umkonfigurierung.