



Doctoral Thesis

Separate connection and functionality is the pivot in embedded system design

Author(s):

Trutmann, Hans Otto

Publication Date:

2000

Permanent Link:

<https://doi.org/10.3929/ethz-a-004044923> →

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](#).

TIK-Schriftenreihe Nr. 39

Hans Otto Trutmann

Separate Connection and Functionality is the Pivot in Embedded System Design

A dissertation submitted to the
Swiss Federal Institute of Technology ETH Zürich
for the degree of Doctor of Technical Sciences

Diss. ETH No. 13891

Prof. Dr. Albert Kündig, examiner
Prof. Dr. Matjaž Colnarič, co-examiner

Examination date: October 23, 2000

© 2000 Institut für Technische Informatik und Kommunikationsnetze TIK, ETH Zürich

ISBN 3-906469-09-3

Abstract

The goal in the development of embedded real-time systems is a solution that satisfies the initial requirements, although this is not enough. There is also a need for appropriate descriptions of the many different aspects of a system, for connecting these representations among each other and with the implementation itself, for easing maintenance and for dealing with organizational chores. All this makes the solution appear as a by-product of a well-organized development process. This development process profits from a clear separation of concerns, which can be achieved with an explicit interconnection model, the architectural pivot specification. Its relative stability allows development to progress independently behind stable syntactic and semantic interfaces.

The three basic activities in control applications, input, processing and output, are dealt with individually within separate problem areas. A problem-oriented approach postulates the use of suitable methods and tools for solving each of these issues, thus embodying different formal representations and explicitly written code on different levels of abstraction. Unison is achieved through generators yielding code with defined generic properties that favor smooth and efficient implementation. The functional problem comprises all the behavioral aspects of a control system, using shared phenomena to interact with the controlled processes in the environment. Solutions to this problem omit any details not pertaining to the relation between the course of events in the environment and the inner states of a model. The connection problem is concerned with information transport from and to the environment. Its solution is based on regular sequential structures.

During development, the common high-level source baseline permits generating implementations for various modes of operation without modifying the abstract descriptions. The approach eliminates manual programming when code alterations are needed to adapt to changes in the hardware environment. It thus allows testing of proposed solutions against numerical models, as well as adapting resources according to emerging requirements any time during the development process. It supports decentralized development work and simplifies target implementations without complicated scheduling, without interrupts, and without the overhead of elaborate operating systems. The code is statically deterministic, and the performance of entire implementations is analyzed exhaustively to assure the desired real-time properties.

Zusammenfassung

Das Ziel bei der Entwicklung eingebetteter Echtzeitsysteme ist eine der Anforderungsspezifikation genügende Lösung; das reicht jedoch noch nicht aus. Es braucht auch den verschiedenen Aspekten angemessene Beschreibungen, Verbindungen dieser Beschreibungen untereinander und zur Implementation, Unterstützung beim Unterhalt, bei der Weiterentwicklung und beim organisatorischen Ablauf. All das lässt die eigentliche Lösung dann eher als die Begleiterscheinung eines gut organisierten Entwicklungsprozesses erscheinen. Dieser Prozess profitiert von klar voneinander abgegrenzten Problembereichen; eine Abgrenzung, die mit der architektonischen Pivot-Beschreibung erzielt wird. Hinter solch syntaktisch und semantisch stabilen Schnittstellen kann innerhalb einzelner Problembereiche selbständig weiterentwickelt werden.

Die drei grundlegenden Aktivitäten in Steueranwendungen, Eingabe, Verarbeitung und Ausgabe, werden gesondert in eigenen Problembereichen behandelt. Ein problembezogener Ansatz postuliert den Gebrauch geeigneter Methoden und Werkzeuge für jedes dieser Gebiete, so dass es verschiedene formale Darstellungen neben explizit geschriebenem Programmtext auf mehreren Abstraktionsstufen gibt. Diese werden mittels Generatoren vereinigt, welche Code mit definierten generischen Eigenschaften liefern, der sich einfach und effizient implementieren lässt. Das funktionale Problem umfasst das Systemverhalten; die Koppelung des Steuersystems mit den zu steuernden Prozessen in der Umgebung läuft über Phänomene, welche von beiden geteilt werden, so dass Lösungen dieses Problems nur die Wechselbeziehung von Ereignisfolgen in der Umgebung mit dem inneren Zustand des Modells beinhalten. Beim Verbindungsproblem geht es um den Transport von Information vom und zum kontrollierten Prozess. Lösungen basieren hier auf sequentiellen Strukturen.

Aus den gemeinsamen abstrakten Beschreibungen lassen sich Implementationen für verschiedenste Betriebsarten generieren, was manuelle Eingriffe beim Austausch von Systemkomponenten überflüssig macht. Die erarbeiteten Lösungen können an numerischen Anlagemodellen ausprobiert werden. Der Ansatz unterstützt die dezentrale Entwicklung von Systemen und vereinfacht die Installation auf dem Zielsystem: keine komplizierte Ablaufplanung, keine Unterbrechungen und kein aufwendiges Betriebssystem. Der erzeugte Code ist statisch deterministisch, und das Zeitverhalten von ganzen Implementationen wird umfassend geprüft, um das gewünschte Echtzeitverhalten nachzuweisen.