Embedded Software Design for Mechatronic Systems *)

Dusko Jovanovic, Gerald H. Hilderink and Jan F. Broenink

Twente Embedded Systems Initiative, Drebbel Institute for Mechatronics,
Control Engineering, Faculty of EE-Math-CS
University of Twente
P.O.Box 217, 7500 AE, Enschede, the Netherlands
Phone: +31 53 489 2788 Fax: +31 53 489 2223
E-mail: D.Jovanovic@utwente.nl

Introduction

This research project is motivated by the fact that nowadays it is impossible to separate control engineering from software engineering. Besides that both of them can be found in definitions of mechatronics, this project deals with exploitation and improvement of their strong natural interdependency.

In all modern reactive systems, what all mechatronics systems are, one will always find one or more embedded computers. The functionality of these computers, and in turn controlled systems, is powered by embedded software [1].

The other important property of reactive systems is the inherent concurrency of their functionality. By notions of modern software design, the natural concurrency in interaction between a reactive system and its environment and within the system itself has to be reflected on the embedded software architecture. Insisting on the resemblance between the real-life problem structure and the structure of the software features important qualities of the software development methodology: proper problem decomposition, solution scalability, separation of concerns among the system components, software and hardware distribution, system components maintainability.

The essential property of computers to be embedded in mechatronics products is that they usually have to be small compared to the size of the product and they must not contribute significantly in the price of the product. These design constraints imply that computer systems with scarce resources are used.

Having together listed (and other) constraints along with demanding (real-time) performance and (functionality) design requirements makes the issues in the field delicate. Especially, modest and heterogeneous hardware resources limit designers’ freedom in choice of software real-time concurrency model and platform.

Approach

The aim of this project is to construct a methodology along with a design tool allowing an evolutionary migration from dynamic modeling of the problem at hand and proper control laws derivation to concurrent software, possibly distributed and retargetable on heterogeneous hardware platforms.

Dealing with concurrency issues is based on the concurrency model of CSP [2] (Communicating Sequential Processes) algebra, supported by the CT [3] (Communicating Threads) libraries for most popular programming languages – thus portable to almost all hardware platforms.

Ongoing work

The concurrency of the software is specified by means of graphical language being developed: CSP diagrams [4]. CSP diagrams are basically block diagrams of communicating concurrent processes. Most actual are experiments with a specific stepwise refinement procedure: to derive the CSP diagrams from block diagrams characteristic for control laws design. In fact, the research strives for a common language during the whole mechatronic design stretch that covers all relevant issues of multi-paradigm modeling, requested by intrinsic multidisciplinarity of design cycle of a typical mechatronic product. On the practical side, the approach fits well with existing modeling and simulation tool 20-SIM, which makes a basis for tool coverage of the design methodology [5].

References


*) This research is supported by PROGRESS, the embedded system research program of the Dutch organization for Scientific Research, NWO, the Dutch Ministry of Economic Affairs an the Technology Foundation STW.