On designing a framework for distributed real-time embedded control systems

Bojan Orlic 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, NL-7500 AE Enschede, the Netherlands
Phone: +31 53 489 2817 Fax: +31 53 489 2223
E-mail: B.Orlic@utwente.nl, J.F.Broenink@utwente.nl

Introduction

Closed loop control systems are in essence parallel and distributed. But when implementing this parallelism in software, lot of obstacles concerning multithreading communication and synchronization issues arise. Using multithreading in safe and structured way is possible if the program is built such that it can be checked using some formal mathematical algebra (e.g. CSP). Fortunately, several formal checking tools and libraries implementing CSP based constructs are available for most widely used general purpose programming languages. One of those libraries is the CT library [1, 2], developed at University of Twente.

2 Approach

CT library was designed with hard real-time embedded systems in mind, and thus it is suitable for implementation of control software of mechatronics systems. This project is about developing communication extension to the CT library to make it applicable in distributed systems connected over fieldbuses [3]. The CAN fieldbus is chosen as first fieldbus because of its deterministic, reliable communications with short prioritized messages and intensive error detection. Besides, its application field has already shifted from automotive networks towards industrial applications.

Many control systems are required to provide a high degree of safety. If such control system temporarily fails for only a brief moment, consequences can be serious. Therefore, the aim is to insure correct work of those systems despite faults. In scope of using the CT library for safety critical systems, a solution is to incorporate a special fault-tolerance protocol layer in this library. The achieved architecture will be tested by artificially producing alarm showers and failures (e.g. node failure, network partition, corrupted, lost messages…).

Apart from achieving reliable multithreading based on CSP and fault tolerance in distributed environment, there are several other orthogonal directions being under investigation in this project: improving real-time bus and processor scheduling, satisfying control-system specific requirements, achieving scalability in distributed environment and achieving flexibility in a level that will not cause conflicting to much with other requirements.

References

[1] G. H. Hilderink, J. F. Broenink, and A. W. P. Bakkers, “Communicating threads for Java,” presented at Proc. 22nd World Occam and Transputer User Group Technical Meeting, Keele, UK, 1999.

[2] G. H. Hilderink, A. W. P. Bakkers, and J. F. Broenink, “A Distributed Real-Time Java System Based on CSP,” presented at The third IEEE International Symposium on Object-Oriented Real-Time Distributed Computing ISORC 2000, Newport Beach, CA, 2000.

[3] B. Orlic and J. F. Broenink, “CSP channels for CAN-bus connected embedded control systems,” presented at Progress 2002 Workshop, Utrecht, 2002.