A REUSABLE AND PLATFORM-INDEPENDENT FRAMEWORK FOR DISTRIBUTED CONTROL SYSTEMS

T.B. Brown[**], T.A. Henzinger[*], C.M. Kirsch[*], A. Pasetti[**], and **W. Pree[**]**

[*]*University of California, Berkeley* - [**]*University of Constance, Germany*
tbb@acm.org, tah@eecs.berkeley.edu, cm@eecs.berkeley.edu, pasetti@fmi.uni-konstanz.de, pree@acm.org

This paper describes the integration of the embedded programming methodology Giotto and the Novato framework to create an environment for the rapid development of distributed software for safety-critical embedded control systems with hard real-time requirements of the kind typically found in aerospace applications.

Giotto is a middleware that offers a tool-supported design methodology for implementing embedded control systems on platforms of possibly distributed sensors, actuators, CPU's and networks. Giotto enables the decoupling of software design from implementation concerns (functionality and timing from scheduling, communication, and mapping). It thus allows developers to concentrate on the design of the software architecture and on the implementation of the control and management functionalities required by the target application. Giotto is based on a time-triggered programming language. This ensures timing predictability and makes it particularly suitable for safety-critical applications with hard real-time constraints. Avionics systems are its natural target applications.

Novato is a software framework for embedded control systems. Software frameworks are a software reuse technology. They consist of collections of components with predefined cooperations among them that capture an architectural design optimized for a specific domain. They predefine the composition and interaction of the components of a system while at the same time allowing for customization by providing hooks where default behavior can be overridden. Frameworks differ from other reuse technologies because they make *architectural* (as opposed to *code*) reuse possible and because they rely on object composition and inheritance as functionality-extension mechanisms. A software framework for satellite control systems was recently developed for the European Space Agency and is now being extended to cover embedded control systems in general.

Giotto and Novato are complementary technologies. The former addresses the physical realization concerns (scheduling, communication, and mapping) of an embedded system whereas the latter addresses the logical correctness concerns (functionality and timing). Their integration creates an environment where real-time embedded control applications can be rapidly instantiated that are guaranteed to be analyzable for their timing properties even when they are distributed over several CPUs. The Giotto-Novato infrastructure allows software components to interact as if they resided within the same address space even when they are located in different processes or in different CPUs. This effect is achieved through the *delegate object* mechanism. This is an innovative approach to distribution that, unlike rival approaches like CORBA or DCOM, is specifically designed to promote timing predictability and is therefore ideally suited for hard real-time applications.

The first part of the paper describes the Giotto and Novato technologies. The second part presents their integration with particular attention to distribution and real-time issues. The third part sketches the application of the integrated Giotto-Novato approach to a motivating example of a satellite control system based on multiple on-board processors. The paper concludes with a discussion of other potential applications to avionics systems.

Timothy B. Brown holds a B.Sc. in Mathematics and a M.Sc. in Computer Science. He has over 12 years experience in various aspects of Object-Oriented development including experience as a programmer, analyst, consultant, team leader, and instructor. He is currently a research assistant in the computer science department of the University of Constance.

Thomas A. Henzinger is professor at the Electrical Engineering and Computer Sciences department of the University of California at Berkeley. He has worked for several years in many areas of formal verification with a recent focus on the design and analysis of embedded real-time systems. He received his Ph.D. in Computer Science from Stanford University in 1991.

Christoph M. Kirsch is postdoctoral researcher at the Electrical Engineering and Computer Sciences department of the University of California at Berkeley. He worked for five years at the Max Planck Institute for Computer Science in Saarbruecken, Germany. He received his diploma degree and Ph.D. in Computer Science from the University of the Saarland in Saarbruecken.

Alessandro Pasetti holds degrees electrical engineering, control systems and sociology. He worked for nine years as a satellite control engineer at ESA-Estec (the main R&D centre of the European Space Agency). He is now a research assistant in the computer science department of the University of Constance.

Wolfgang Pree is professor of computer science at the University of Constance. He has worked for several years in many areas of software engineering, in particular focusing on object-oriented software development, software architectures, frameworks, and human-computer interaction. Wolfgang is the author of Design Patterns for Object-Oriented Software Development (Addison-Wesley, 1995).