Embedded system design using Polychrony

Espresso Team, IRISA
Campus universitaire de Beaulieu
35042 Rennes Cedex
http://www.irisa.fr/espresso/welcome_english.html

1. Polychony Overview

Polychrony is an integrated development environment and technology demonstrator consisting of a compiler, of a visual editor and of a model checker. It provides a unified model-driven environment to perform embedded system design exploration by using top-down and bottom-up design methodologies formally supported by design model transformations from specification to implementation and from synchrony to asynchrony.
Polychrony supports the synchronous, multi-clocked, dataflow specification language Signal [2] [3]. It allows to perform validation and verification tasks, with the integrated Sigali model checker [4][9].

Polychrony is developed by the project-team Espresso, whose scientific objectives are to define and implement models, methods and tools for the computer-aided engineering of trusted application architectures in embedded and mission-critical systems [5]. Typical domains include:

* Process control,
* Signal processing systems,
* Avionics,
* Automotive control, Vehicle control systems,
* Nuclear power control systems,
* Defense systems, Radar systems,

2. The POLYCHRONY workbench

The POLYCHRONY toolset is freely distributed (at binary level) for non-commercial use on the site [1].

Based on the SIGNAL language, it provides a formal framework:

- to validate a design at different levels,
- to refine descriptions in a top-down approach,
- to abstract properties needed for black-box composition,
- to assemble predefined components (bottom-up with COTS).

It constitutes a development environment for critical systems, from abstract specification until deployment on distributed systems. It relies on the application of formal methods, allowed by the representation of a system, at the different steps of its development, in the SIGNAL polychronous semantic model.

Polychrony toolset is a set of tools composed of:

- A SIGNAL batch compiler providing a set of functionalities viewed as a set of services for, e.g., program transformations, optimizations, formal verification, abstraction, separate compilation, mapping, code generation, (C, C++, Java), simulation, temporal profiling...These transformations can be applied or not according to the objective of the compiling.
- A graphical user interface with interactive access to compiling functionalities.
- The SIGALI tool, an associated formal system for formal verification and controller synthesis [4][9].

The company TNI-Valiosys [6] supplies its commercial implementation, RT-Builder, used for industrial scale projects by Snecma/Hispano-Suiza and Airbus Industries.

3. Avionics application modeling with Polychrony

The APEX interface, defined in the ARINC standard [7], provides an avionics application software with the set of basic services to access the operating-system and other system-specific resources. Its definition relies on the Integrated Modular Avionics approach (IMA [8]). A main feature in an IMA architecture is that several avionics applications (possibly with different critical levels) can be hosted on a single, shared computer system. Of course, a critical issue is to ensure safe allocation of shared computer resources in order to prevent fault propagations from one hosted application to

another. This is addressed through a functional partitioning of the applications with respect to available time and memory resources. The allocation unit that results from this decomposition is the partition (Figure 1).

A partition is composed of processes which represent the executive units (an ARINC partition/process is akin to a UNIX process/task). When a partition is activated, its owned processes run concurrently to perform the functions associated with the partition. The process scheduling policy is priority preemptive. Each partition is allocated to a processor for a fixed time window within a major time frame maintained by the operating system. Suitable mechanisms and devices are provided for communication and synchronization between processes (e.g. buffer, event, semaphore) and partitions (e.g. ports and channels).

The specification of the ARINC 651-653 services in SIGNAL offers a complete implementation of the APEX communication, synchronization, process management and partitioning services. Its SIGNAL implementation consists of a library of generic, parameterizable SIGNAL modules.

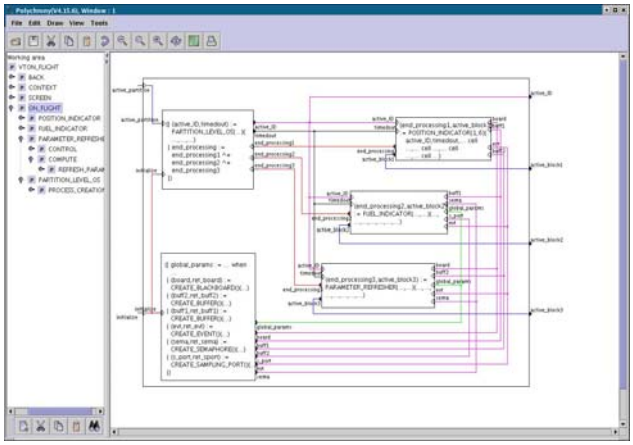

Fig. 1 : Avionics application modeling using the visual editor of the Polychrony workbench

4. References

[1] http://www.irisa.fr/espresso/Polychrony

[2] Jean-Pierre Talpin and Paul Le Guernic. **An algebraic theory for behavioral modeling and protocol synthesis in system design**. In Formal Methods in System Design, Special Issue on formal methods for GALS design, Kluwer Academic Publishers, 2005 (to appear).

[3] Paul Le Guernic, Jean-Pierre Talpin, and Jean-Christophe Le Lann. **Polychrony for system design**. Journal for Circuits, Systems and Computers, Special Issue on Application Specific Hardware Design, World Scientific, April 2003.

[4] http://www.irisa.fr/vertecs

[5] Abdoulaye Gamatié, Thierry Gautier, and Loïc Besnard. **Modeling of Avionics Applications and Performance Evaluation Techniques using the Synchronous Language SIGNAL**. In SLAP 2003 Porto, Portugal, July 2003.

[6] http://www.tni-valiosys.com

[7] **Design Guidance for Integrated Modular Avionics**, Technical report, ARINC Specification 651-1, Airlines Electronic Engineering Committee, November 1997.

[8] **Avionics Application Software Standard Interface**. Technical report, ARINC Specification 653, Airlines Electronic Engineering Committee, January 1997.

[9] H. Marchand, P. Bournai, M. Le Borgne, P. Le Guernic, **Synthesis of Discrete-Event Controllers based on the Signal Environment,** *Discrete Event Dynamic System: Theory and Applications,* 10(4):325-346, October 2000.