January 13-17, 2014
Computer Science Department, Research school ER02
Contact : Laure.Gonnord@ens-lyon.fr

ENS DE LYON

Programming Embedded Systems with synchronous langages

Summary

This research school aims at introducing the synchronous paradigm, from its first definition in the early 80's to its last developments. Historically, the first synchronous languages were the first attempt in introducing time into the development of (reactive) embedded systems, that is, systems that interact with their environment.

Typically, a reactive system (a fight control system of an airplane for example) gets some stimuli from its sensors, and according to these values compute the new orders to its activators, and repeat this infinitely. Under some hypothesis, (real) time can be abstracted away and the programmer can only program the computation itself (the inner code of the infinite loop). Synchronous languages provide the notion of this logical time and primitives to program reative systems.

The school will explore numerous aspects of synchronous programming, from language to verification, including high-level specification and verification, sequential and parallel/distributed code generation, hardware low-level description, and real-time programming. The data-flow language Lustre, as well as other implementations such as Esterel, Prelude, will be described.

During the week, the different speakers will demonstrate that far from being an "old" paradigm, synchronous programming has shown its applicability and is still accurate for numerous applications, from low-level hardware description and verification to high level real-time system development. Moreover there are still a lot of research directions, mainly in introducing some of the synchronous ideas in more "traditional" languages.

1 Schedule

— Monday, 9-12 14-17 : Nicolas Halbwachs, CNRS senior researcher, VERIMAG, Grenoble, France

— Tuesday 9-12 14-16 : Nicolas Halbwachs

— Wednesday 9-12 14-17 : Abdoulaye Gamatié, CNRS researcher, LIRMM, Montpellier, France

— Thursday 9-12 : Alain Girault, Inria Grenoble senior researcher, France

— Friday 9-12 14-17 : Julien Forget, Assistant Professor University of Lille, France

2 Content

2.1 Monday 13 , Tuesday 14 : General introduction, Lustre and Esterel, compilation and verification

1. Introduction : Reactive systems - Usual implementation methods - The synchronous point of view.

2. The Data-flow language Lustre : The data-flow approach - Lustre, the combinational part, temporal operators, clocks - The story of Lustre/Scade
3. The imperative language Esterel : Introductory example - Signals, events occurrences - The kernel language - Derived statements - The story of Esterel - SyncCharts
4. Compilation of synchronous languages : Static semantics, causality analysis - Sequential code generation, single loop - Explicit control automaton - Implicit automaton, from Esterel to Lustre - Reincarnation in Esterel
5. Verification and automatic testing : Synchronous observers - Verification by model-checking - Numerical properties - Automatic testing
6. Mini-seminars (if there remains time) - Arrays in Lustre V4 and V6 - Use of synchronous languages for modeling complex systems

2.2 Wednesday 15 : Abstract clock-based approaches in the programming, design and analysis of embedded systems

1. Multi-clock system programming with the Signal language (language constructs, static analysis and code generation)
2. An introduction to the CCSL formalism
3. Design space exploration for Multiprocessor Systems-on-Chip using abstract clocks
4. Practical session on Signal programming in the afternoon.

2.3 Thursday 16 Morning : Parallel / Distributed Synchronous Systems

1. Globally asynchronous locally synchronous systems (GALS)
2. Automatic distribution of Lustre and Exterel programs : from the explicit automata – from the implicit automata – from control points.
3. Long-duration tasks and synchronous model.
4. Modular distribution of Heptagon programs
5. GALS programming languages : SystemJ and DSystemJ.

2.4 Thursday 17 : Logical time and real-time in the Synchronous approach

1. Multirate system Design
2. Synchronous Real-Time.
3. Prelude : language and compilation.
4. Practical session on real-time programming in the afternoon.