Office National d'Études
et de Recherches Aérospatiales

Centre d'Études et de Recherche de l'École Nationale Supérieure
de l'Aéronautique et de l'Espace à Toulouse

SATURNE: a Reactive - Anytime Programming Model for Intelligent Embedded Real-Time Systems

(short paper)

Martin Adelantado **Frédéric Boniol** **Simon de Givry**

Publication présentée à :

3rd IEEE Workshop on Parallel and Distributed Real-Time Systems
Santa Barbara, California, April 24-26, 1995.

Département d'Études et de Recherches en Informatique

2, avenue Édouard Belin - BP 4025 - 31055 Toulouse Cedex
Tél. 62.25.25.91 - Fax. 62.25.25.93

SATURNE: a Reactive - Anytime Programming Model for Intelligent Embedded Real-Time Systems *

M. Adelantado F. Boniol S. de Givry

CERT-ONERA, Dépt. d'Informatique
2, av. E. Belin, 31055 Toulouse cedex, France
saturne@cert.fr

1 Introduction

One of the major challenge of next embedded systems is to involve *intelligence* while preserving the classical real-time properties: (a) *reactivity, i.e.* the capability to react continuously towards asynchronous inputs, and (b) *predictability*. Furthermore, one of the main property which is requested in this kind of systems operating in a highly non-deterministic outside environment, is *adaptability* to new and unexpected conditions, and particularly to dynamic temporal deadlines.

Strong predictability often means that it should be possible to prove both the functional and temporal behavior of a system, in response to any combination of signals from outside environment, and particularly to prove that critical tasks meet their deadlines. On the contrary, in non-deterministic environments involving dynamic timing constraints, the assumption that critical tasks meet their deadlines can not be guaranteed. In that sense, strong predictability and adaptability are contradictory requirements [1]. Furthermore, intelligence often means complex computations. In that sense reactivity and punctuality seem also to be incompatible with reasoning capabilities.

The aim of this paper is to investigate a programming model, called SATURNE, addressing the issue of intelligent real-time systems, that is how to mix adaptability, reactivity and predictability. We only focuses on adaptability to temporal deadlines, addressing the problem of guaranteeing a response when temporal deadlines may be statically unpredictable. SATURNE is based on a mixed *reactive-anytime* approach. The main idea is to introduce firstly a reactive model of computation providing predictability, and secondly an anytime model of computation providing adaptability to dynamic temporal deadlines.

2 The SATURNE Model: Overview

SATURNE is based on the classical dichotomy proposed by D. Harel and A. Pnueli between *transformational* and *reactive* systems [5]. From this point of view, a real time system is composed of two layers: (a) a set of transformational tasks that perform classical computations, and (b) a reactive subsystem that handles inputs and controls the application. From that point of view, reactivity and predictability are ensured by the reactive part. Its behavior must be then deterministic and temporally predictable.

Several solutions have been proposed for programming pure reactive systems: *asynchronous* models such as CSP, Petri Nets ..., or *strong synchronous* languages such as Esterel, Lustre, Signal [2]. However, due to a non-deterministic model of concurrency, asynchronous models lead generally to concurrent systems that are non-predictable. On the other hand, strong synchronous languages consider ideal reactive systems that produce outputs synchronously with inputs. From an implementation point of view, such a semantics provides a powerful manner to produce predictable centralized reactive processes. But it is rather more unrealistic in the area of distributed applications where communication times cannot be neglected.

The approach investigated by SATURNE is based on a *strong-weak synchronous* paradigm [4]. The reactive part of a SATURNE application is composed of a set of strong synchronous kernels communicating in a weak synchronous way: kernels are synchronous with a global clock, *i.e.* firstly they react simultaneously at each "tick" of this clock to signals coming from outside environment or broadcast by other kernels at previous tick, and secondly they broadcast their output signals to other kernels for the next tick. In that sense, the system operates exactly the same as a parallel composition of abstract machines in which the next state is computed instantaneously and in which changing state and broadcasting outputs require waiting for the next clock tick. Such a computation model ensures reactivity and predictability.

Furthermore, adaptability to dynamic temporal deadlines means that the system must be able to produce a response to unpredictable deadlines, and then *at any time*. This leads to the *anytime* model of computation for transformational tasks [3, 6]: by definition an anytime algorithm is able to produce a partial or reduced quality output at anytime before its completion. Consequently, reactive kernels can interrogate such transformational tasks at any time according to temporal deadlines.

SATURNE applications are then composed of con-

*This research is supported by the French Ministry of Defence (DRET).

current *reactive/anytime* modules communicating in a weak synchronous way. Each module involves a strong synchronous reactive kernel, controlling anytime emergency deliberative tasks and conventional periodic processes. The reactive kernel contains the logical and temporal control, activates and stops the tasks according to the deadlines and to the current situation.

3 Reactive Kernels

SATURNE Reactive kernels are written in Esterel synchronous language, extended by a set of statements for controlling anytime tasks. Esterel is a formal *signal-calculus*, allowing description of causality relations between input and output signals. An Esterel kernel reacts to input signals, assumed to be present at this time, by executing its body, and broadcasts corresponding output signals. This reaction is assumed to be deterministic: for any state of the program and any set of input signals, there is exactly one possible set of output signals.

In order to control anytime tasks, we have extended Esterel by a set of synchronous control directives including: (a) activating an anytime agent (`start`), (b) consulting the current response of an anytime agent (`consult`), (c) killing an anytime agent (`kill`), (d) suspending an anytime agent (`suspend`), and (e) resuming an anytime agent (`resume`). Finally, with each anytime agent is associated an implicit signal called `END`_*task_name* automatically sent by the agent as soon it has completed.

Such a strong synchronous approach for programming reactive kernels of SATURNE modules is interesting in that it permits to tackle real-time and reactivity in a formal and deterministic way. Secondly, extended Esterel programs are compiled into deterministic sequential automata, yielding excellent run-time efficiency, and allowing formal analysis using verification tools.

4 Deliberative Tasks

To achieve adaptability to timing constraints, SATURNE involves an *anytime* model of computation for emergency deliberative tasks. The concept of anytime computing was introduced independently by M. Dean and Th. Boddy [3], and by R.J. Horvitz who refers to this model as flexible computations [6]. These approaches assume that a response is guaranteed for any allocation of computation time and that the corresponding computations are expected to return better responses when given more time. This model is based on the informal idea that for critical real-time applications, it is often better to return a crude and incomplete solution in time than one more detailed and fully elaborated but too late.

Deliberative tasks are controlled by reactive kernels by using the directives described above. The behavior of these tasks at run-time may be described as follows: (a) initially, the task is idle and becomes running after receiving a `start` command; (b) afterwards, the task is waiting for a `kill` or a `suspend` command, or can completed by producing an `END` signal; (c) finally, whether the task is running, suspended or terminated, it can be consulted at any time, without changing its current state.

The anytime model of computation introduces several advantages in meeting timing constraints. Firstly, it allows to face dynamic temporal deadlines. Secondly, it relaxes the scheduling problem. In particular, the correctness of scheduling strategy is no more a binary function since a response can be guaranteed even when processes are interrupted before their completion.

5 Conclusion and Future Trends

The SATURNE model presented in this paper investigates a framework for designing real-time systems requiring predictability, reactivity and adaptability to timing constraints. SATURNE is based on the assumption that real-time systems can be built around reactive kernels containing timing constraints, conventional periodic processes, and non-reactive anytime tasks facing unpredictable situations.

This model has been experimented on an actual military assistant system, the "Pilot's Associate" of Dassault-Aviation French Company, involving mainly 4 modules: sensor fusion, tactical assessment, mission assessment, and tactical management. The SATURNE implementation of this application involves 8 reactive kernels, 17 conventional transformational tasks and 6 anytime emergency tasks.

Future trends include two main directions. The first one is to formalize the anytime model of computation from a programming point of view. The second one is to address the problem of fault-tolerance in such a context.

References

[1] M. Adelantado, F. Boniol, V. David, B. Lécussan, and R. Porche. "Predictability in Distributed Intelligent Real-Time Systems", *First IEEE Workshop on Parallel and Distributed Real-Time Systems*, Newport Beach, California, April 1993.

[2] A. Benveniste and G. Berry. "The synchronous approach to reactive and real-time systems", *Proceedings of IEEE, Another Look at Real-time programming*, 79(9):1270–1282, September 1991.

[3] M. Boddy and Th. Dean. "Solving Time Dependent Planning Problem", *Eleventh IJCAI-AAAI Conference*, August 1989.

[4] F. Boniol and M. Adelantado. "Programming Distributed Reactive Systems: a Strong-Weak Synchronous Coupling", *7th International Workshop on Distributed Algorithms*, LNCS 725, Lausanne, Switzerland, September 1993.

[5] D. Harel and A. Pnueli. "On the development of reactive systems", *Logic and Models of Concurrent Systems*, NATO ASI Series F vol. 13, 1985.

[6] R.J. Horvitz. "Reasoning about beliefs and actions under computation resource constraints", *Workshop on Uncertainty in Artificial Intelligence*, Seattle, Washington, 1987.