Chapter 2

UML for Real-Time

Which native concepts to use?

Sébastien Gérard and Francois Terrier

CEA-LIST (“Commissariat à l’Energie Atomique - Laboratoire d’Intégration des Systèmes et des Technologies” is the Systems and Technologies Integration Laboratory of the French Agency for Atomic Energy)

Abstract: Engineers are increasingly facing the hard problem of developing more sophisticated real-time systems while time to market and cost constraints are constantly growing. The necessity of adopting object oriented modeling in the real-time domain appears to be essential in order to face the rapidly changing market conditions. The main obstacles are the lack of standards and the mismatch with real-time needs. With the standardization of UML, the first main drawback is being reduced. Current work performed at the OMG on UML standards evolution to better integrate real-time issues shows both that there is a strong interest in the subject and that current proposals are neither completely satisfying, nor completely compatible. This chapter aims to describe in minute detail what UML proposes as support for parallelism, behavior and communication modeling, and how it is also possible to express quantitative real-time features (such as deadlines, periods, priorities...). Apart from UML, OMG has specified two additional profiles well-suited for real-time, the Scheduling, Performance and Time profile and the Action Semantics profile. Due to size limitations, the goal of this chapter is not to describe precisely their content. It aims at skimming through both profiles to outline their purpose and content. Finally after having detailed the native possibilities of the UML in terms of notations for real-time, this chapter outlines a prospective approach showing how to use such notations to build real-time applications.

Key words: UML, real-time, concurrency, communication, behavior

1. INTRODUCTION

Several years ago the real time and embedded systems market was considered as a very specific and “confidential” sector (about 5% of the
global market of software based systems). Now several studies of the evolution of the software market consider that embedded systems may represent by the year 2003 and beyond more than 50% of the global market (including personal computers, client-server and information system applications).

This explosion of the market for real time systems and the constant increase of embedded services force engineers to face more and more the hard problem of developing sophisticated real-time systems while competition through time to market and cost constraints increases day after day. Classical real-time development of software systems is reaching its limits in a world where on the one hand hardware targets cannot be known in advance and where on the other hand version evolution becomes increasingly fast and time to market must be shortened drastically in order to meet economic requirements. Reusability and evolvability become mandatory requirements for modeling methods and development techniques. In such a context, real-time systems development cannot be achieved efficiently without a strong methodological support and accompanying tools. In parallel, object oriented techniques have reached a sufficient level of maturity to successfully provide the flexibility required by new applications. Up to now however, the real-time community has been long reluctant to cross this Rubicon; mainly for both of the following reasons:

- The state of object-oriented approaches was not mature enough to provide stability in their solutions (methods, tools,...);
- The real-time specifics were generally not well covered by existing methods.

During the last few years UML has become the lingua franca among system modelers all over the world. With the arrival of the UML standard in 1997, the signal that many editors were waiting for has appeared, and a first step is being achieved that will permit the spread of a new tool generation. Indeed, object oriented modeling with a standard formalism such as UML brings significant solutions to the issues mentioned previously. At least, UML is becoming a “de facto” widespread standard for software engineering, and also object oriented approaches now fit quite well both of the following needs: (i) to have a fine level of modularity for component based development; and (ii) to improve reusability and maintainability properties of subsystems.

For UML and real-time, the OMG Platform Technology Committee is concerned with both task forces, the Analysis and Design PTF\(^\text{11}\) (AD PTF - http://adtf.omg.org/) and the Realtime, Embedded, and Specialized Systems Platform Task Forces are groups dedicated to a domain that may initiate Requests For Proposals (RFP) in order to later produce new standards.