

Towards efficient RT systems modeling with the AIT-WOODDES approach

Pontus Jansson, Mecel AB, Gothenburg, Sweden, pontus.jansson@mecel.se
Sébastien Gérard, LETI-DEIN - CEA/Saclay, Gif sur Yvette Cedex, France,
Sebastien.Gerard@cea.fr

Abstract

SW development for complex distributed real-time systems places great demands on development process, methodology and CASE tools. The AIT-WOODDES project aims to contribute to improve standards used in the development of real-time embedded systems by providing development methodology, modeling language, and a development tool platform. Automotive manufacturers and their subcontractors will benefit from model-based development, i.e. be able to truly share and work with the same specification models on the same level of abstraction.

AIT-WOODDES aims to demonstrate that real-time embedded systems can be completely defined using the UML standard [UML] and object oriented methodology. The XMI standard for model exchange will be used to enable tool platform interoperability. Verification and validation concerns will be addressed by adapting available techniques such as formal verification and automatic test case generation.

The AIT-WOODDES methodology and tool platform will support the new trend to reform OEM-Subcontractor relationships in order to improve the development process. This approach will have impact on development time, cost and product quality and could provide the Delphi Engineering Organization with valuable knowledge. Adopting the project approach will have great potential to increase competitiveness in the automotive software market.

Current work on requirement analysis methodology is showing promising results to integrate the AIT-WOODDES methodology and UML profile with full tool support in the development process. Both methods and tools support the possibility for OEMs and subcontractors to adopt roles and take responsibility for development according to their own ambitions and experience.

Project Overview

The AIT-WOODDES project "Advanced Information Technology - Workshop for Object Oriented Design and Development of Embedded Systems" [WOODDES] is part of the EC funded Information Society Technologies [IST] research programme.

Participants

The project has participants from both automotive and telecom industry, academics and leading tool vendors. Mecel AB is participating on behalf of Delphi Delco Electronics and has the responsibility for the case study development and assessment.

- End users
 - PSA (Peugeot Citroën Automobiles), France: Automotive OEM
 - Mecel AB (Delphi), Sweden: Automotive subcontractor
 - INTRACOM, Greece: Telecom service and system developer
- Laboratories/academics
 - CEA-LETI, France: technology provider for real-time development with UML (ACCORD) and for automatic test case generation (AGATHA tool)
 - University of Uppsala, Sweden and Aalborg University, Denmark: technology provider for model checking of timed automata (UPPAAL tool)
 - OFFIS, Germany: technology provider for formal model checking (Model Checker tool)
- Tool vendors
 - I-Logix, Israel: editor and vendor of a UML-RT development tool (Rhapsody)
 - Softeam, France: editor and vendor of a UML modeling tool (Objecteering)

Objectives and benefits

The AIT-WOODDES project will provide methodology, tools and contribute to standards used in the development of real-time embedded systems. The focus will be set on process continuity between the industrial participants through international standards and interoperability of tools. Object oriented methods will contribute to component-based software engineering providing homogenous support for the development process. Validation concerns will be addressed by adapting available techniques and tools. AIT-WOODDES will demonstrate that real-time embedded systems can be completely defined using the UML standard (Unified Modeling Language from the Object Management Group, OMG). AIT-WOODDES will significantly increase competitiveness in this emerging market by permitting reuse of components and closer co-operation between industrial partners.

The results of the project are expected to have major impacts on:

- development time and cost
- quality of delivered products
- continuity of the development process

To be competitive, companies need to decrease time to market. To achieve this, the project proposes to optimize the use of different actors' know-how in design and development techniques:

- to investigate and adapt the UML notations towards a development process for embedded systems with real-time constraints.
- to implement homogeneous modeling tools for data exchange between people from different areas using different data administration systems
- to extend validation tools to be used early in the process (during the prototyping phase) and to cover both component validation and integration.

Benefits for end-users will be a new object-oriented development process that considerably reduces the costs via:

- decreased design, prototyping and validation time since the users can easily derive from the UML high-level model many different specific models;
- decreased feasibility and analysis time;

- increased product quality through better reliability and safety, earlier validation of the specifications and reuse of successful components.

Benefits for tool providers will be an integrated design toolset that takes as input UML models, validates the system design and automatically generates the executable model (i.e. the target code). The approach developed in the project should result in a proposal for UML that solves many deficiencies of the current notations, especially the lack of formal semantics. Project partners are already involved in OMG standardization activities, and based on the project results they will propose UML standard improvements totally compatible with the project solutions.

Organization

WP0: Project management

The main role of the project management is to provide the necessary co-ordination among all the partners in the consortium and to ensure a common understanding of the project objectives.

WP1: Common methodology

- Identification of real-time concepts that must be expressed in the specifications of real-time embedded systems. Particular attention is given to requirements used and expressed, respectively, by the final systems end users and by systems developers.
- Definition, formalization and standardization of a UML profile dedicated to real-time embedded systems modeling and allowing model exchange among different actors during all the different modeling stages.
- Definition of a common methodology that will provide end users with both formalisms and modeling tools guidelines
- Methodology integration by end users that will be a first step for them to appropriate AIT-WOODDES results (as methodology and the use of the profile characteristics), both along their whole development cycle and among all actors.

WP2: Tools interaction mechanisms

This workpackage aim at defining the user environment and the tools that will support the AIT-WOODDES process as defined in WP1.

- Definition of the tool platform that will support the process and the methodology based on UML for the design of real-time systems, providing basic mechanisms to exchange models between the tools.
- Extension and adaptation of the tools for supporting the file export/import of UML extended models by means of a textual format based on XML.

WP3: Validation of real-time systems

The workpackage is concerned with providing support for formal verification techniques to reduce the validation effort through automatic or aided validation tools. In complement, early simulation and back animation of the model from the application execution are efficient tools when validating the principle of a system or to aid its debugging. And, finally automatic test generation will both reduce the test phase cost and increase the confidence of the embedded systems implementation.

WP4: Case studies

The main objectives of the case study are to:

- Validate the AIT-WOODDES methodology by application modelling, model exchanging, prototyping implementation and validation at all stages of the development.
- Assess the use of UML in the AIT-WOODDES development methodology for real-time embedded systems.

The applicability and benefits of the project approach, i.e. the design framework and the tools that support it will be evaluated during the application case studies. The evaluation process refers to all steps of the development cycle including analysis, simulation, design, implementation / code generation, verification and validation of the chosen applications.

WP5: Exploitation and dissemination

This workpackage will be in charge of introducing the new technology for real-time embedded systems design defined in the project. The following key tasks are defined:

- define the dissemination plan for the project and exploitation plans for each participant, based on the results achieved in the project, the technology positioning, the market forecast and the investment required for industrialization
- provide the supporting tools to the two industries represented in the project
- transfer the new results especially the extensions to UML for real-time, to the standardisation bodies at the OMG, and make the results known to other interested standardisation groups

Key Concepts

Methodology & Profile

One of the project objectives is the definition of a satisfying solution to the problem of model continuity and exchange between the different actors. One part of this solution is the definition of a modeling methodology based on a formalism that provides real-time concepts.

The approach proposed by AIT-WOODDES is based on the modeling formalism UML. The UML diagrams constitute a specification model that can be used along each step of the system life cycle by the various actors involved. The reasons for this choice are based on the fact that UML is the de facto standard within the SW community and supported by an increasing number of CASE-tools. Even though UML is considered too general to be used for specifying real-time systems and lacks the basic concepts needed to deal with specific real-time concepts, it has the necessary mechanisms that enables an extension that will meet the requirements on real-timeliness. The AIT-WOODDES UML profile will extend the UML notation to allow high level expression of real-time constraints and properties such as timing, periods of frequency, quality of service, etc. Until now, this kind of information appears mainly either as textual comments on analysis models or through final implementation concepts such as task priority [SELIC94, ARTHAUD95, AWAD96, DOUGLASS99, MOORE00].

Model-based development

Real-time requirements and the inherently complex nature of modern automotive embedded systems increases the difficulty to develop and maintain them and requires new efficient methods in order to keep the design focus on the industry skill instead of developing specific competence in real-time and embedded software implementation.

Automotive manufacturers and their subcontractors will benefit from model-based development, i.e. be able to truly share and work with the same specification models on the same level of abstraction. A clear definition of the role of the different actors of the development cycle is required for information/model exchange facilities. This development process requires efficient facilities to exchange the evolving system model among the teams, to ensure the model continuity and its understanding by actors with various technical background and objective.

According to the AIT-WOODDES approach, modeling of a real-time application is essentially based on three models that are in themselves partial, but consistent and complementary views of a larger, global model (Figure 1).

The general idea behind application development is to always manipulate the same model, thus constantly and iteratively refining it at each phase in the modeling process. The use of a single formalism, i.e. UML, all along the development cycle will provide continuous and uniform development. Partial models contributing to the global model are as follows:

Structural model

This model defines the general architecture (topology) of the application in terms of classes and the relations between them. Its modeling function is limited to the "class" level of the

application, i.e. to specifying both local class properties and those affecting the application as a whole.

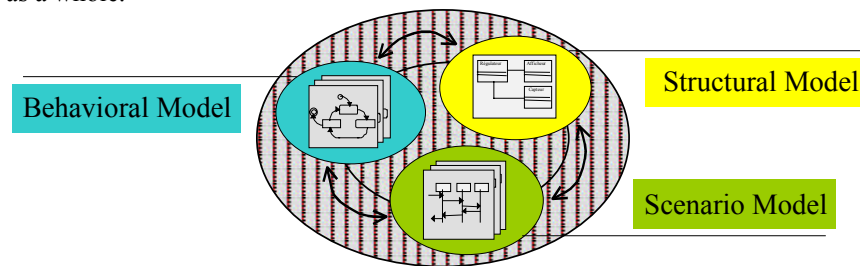


Figure 1 The Global Model

Behavior model

This model defines the behavior of classes involved in the application. It is likewise concerned with the "class" level only, i.e. with specifying class behavior. The behavioral model introduces two object views: that of the protocol, which specifies the global behavior of the object; and the "triggering" view, which accounts for reactive behavior of objects such as reactions to received signals, periodic behavior, etc.

Scenario model

This model is concerned with application "instances" and defines message passing between these various instances for the purpose of performing a given task. The interaction model is specifically described by UML use case and sequence diagrams.

The Real-Time Active Object concept

Support for real-time development in AIT-WOODDES is based on the use of an object oriented approach including a new paradigm in the modeling process: the Real-Time Active Object concept. This paradigm is an extension of the active object concept introduced several years ago in research on concurrent programming languages [YONE97]. One of its first implementations in a development environment is the actor concept of ROOM. The Real-Time Active Object will benefit from the active object capabilities to encapsulate all control mechanisms needed to ensure data consistency and data access synchronization. The AIT-WOODDES approach, concerning the real-time active object together with the structured use of UML statecharts, is based on the work done at the Commissariat à l'Energie Atomique (CEA -LETI) in France by Pr. François Terrier, Dr Sébastien Gérard et al. [TERRIER96, ACCORD98, GERARD00, RTS2000]

Model exchange and OMG standards

These innovation objectives will be completely reached only if the models developed conforming to this UML profile can be exchanged with existing UML modeling tools dedicated to real-time development. For that, the project will develop interchange formats and procedures to allow tool interoperability without missing model information or semantics. The OMG is defining a textual standard (XMI) for UML model representation that will greatly facilitate model exchange among tools supporting this standard. However, this will not solve all the exchange problems, because the basic approach to model and manage real-time features of each tool can differ significantly. Model interpretation or transformation may be necessary to pass a model from one tool to another. The use of continuous models along the whole development process that are based on a common adopted standard and the possibility to exchange models among the several teams and tools involved in the process will increase quality and reduce the costs.

At the OMG the Analysis and Design Task Force monitors the continuous development of UML. Work in progress relating to UML includes the definition of a "Stream-based Model Interchange Format" for the export/import of UML models, and the definition of an "Action Semantics" for the state-chart transitions and the operations in class diagrams. The AIT-WOODDES consortium (e.g., CEA, I-Logix and Softeam are already voting members of OMG) will actively participate with the working groups at OMG on the related RFPs, to ensure the convergence and compatibility of the solutions it develops. It will also make proposals for integration in the new UML standards of its requirements that would not be covered by the other proposals. AIT-WOODDES will directly contribute to these RFPs by

proposing its real-time extensions for UML, by consolidating the action semantics to cover these timing aspects in order to perform simulation, and by completing the capability for exchange of extended UML models between the support tools. It will make an original proposal led by automotive and telecom industrials.

Validation of real-time embedded systems

A critical key point for embedded real-time systems development is their validation. Normally validation appears only at the final product stage. However, it is clear that the earlier the detection of a specification error is, the cheaper it will be to fix. Therefore, the project must provide validation support from the first stage of the development down to the final release of the product.

When looking at testing and simulation, not even considering the real-time features, UML is still missing complete action semantics that prevent CASE tool makers from developing simulators, checkers, verifiers or code generators as it is possible for a formal language such as SDL [SDL93]. The only solution is to interpret the ambiguous or missing semantics of UML constructs, with the risk to be incompatible between tools or to be in conflict with a coming OMG resolution. A formal semantics of UML models, developed in conformity with the UML profile and method proposed by the project, must be provided in order for the tools to offer consistent support for checking, simulation and verification. In addition, the project initially targets the integration of existing validation techniques into its UML modeling framework in order to provide usable solutions during the project. This will address the following points; adaptation of formal analysis techniques applicable to UML models, model simulation, and test generation from the models. Validation model and results will be formalized with UML notations, so they can be exchanged from one stage to another among the several teams along the development process.

Current status

The project is now in its second year and the assessment activities have started. We are currently performing the analysis phase of the case study and the analysis methodology is being revised.

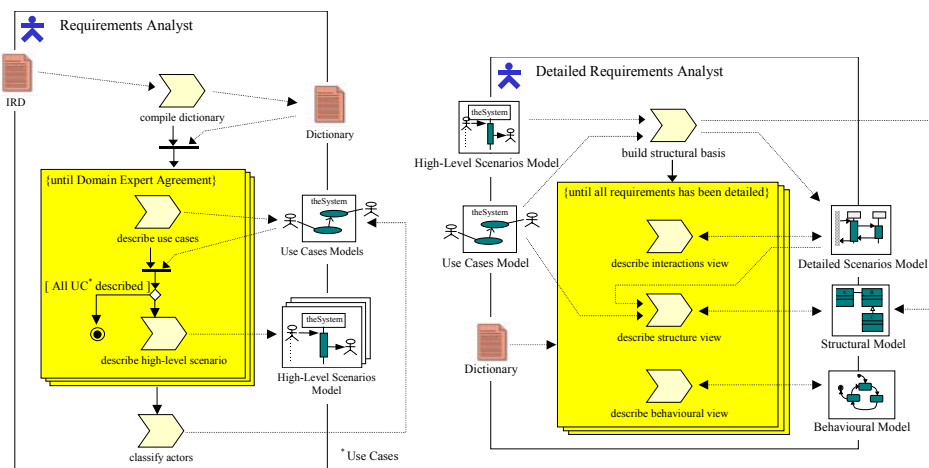


Figure 2 Preliminary and detailed analysis activities

There are two important activities during this phase namely, preliminary and detailed analysis modeling, Figure 2. Preliminary analysis modeling is concerned with specifying the overall functions of the application, in very general terms, as well as the interactions with the environment. Detailed Analysis Modeling then provides as thorough and accurate as possible an assessment of the functions to be performed by the system.

A central issue in this work is the concept of interfaces. The case study participants are working on a scenario where Mecel as a subcontractor is developing a subsystem to an application developed by PSA as the OEM. In this scenario it is important for the actors to be able to specify the interfaces unambiguously and that the different modeling tools support the concepts.

At the analysis stage, in addition to various structural and functional aspects, the designer specifies the real-time behavior of the application. This is where the WOODDES methods and UML profile concepts are utilized to model the constraints pertaining to quantitative real-time features (e.g.: deadlines, periods, etc.) and qualitative one (e.g.: concurrency, parallelism, etc.). The core of these concepts is outlined briefly below.

Dynamic behavior specification

Regarding the behavioral aspects of an application specification, the AIT-WOODDES approach relies on both salient tenets: first, the Real-Time Active Object which is formally defined within the AIT-WOODDES UML profile defines; and secondly, it promotes a structured way to model dynamic behavior with statechart diagrams [DIPES2000].

A real-time active object has four parts: operations, attributes, a mailbox and a local controller as illustrated in Figure 3.

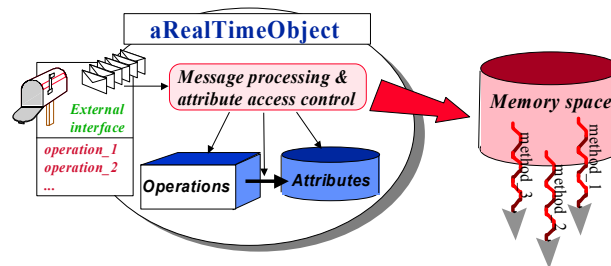


Figure 3 The Real-Time Active Object

A real-time active object is able to receive messages in its mailbox awaking its local controller. The latter, checks the timing constraint attached to incoming messages and dequeues one message following a selection algorithm that selects the message regarding its real-time constraint (e.g. deadline or priority). The local controller then verifies the concurrency constraints with the already running methods of the real-time active object and allocates, when possible, a new thread to handle the new incoming message. When a method terminates its execution, the corresponding thread is released and concurrency constraints are relaxed. If the service is periodic, the thread is not released and supports all periodic executions of the requested service. The local controller is thus in charge of mailbox management, local scheduling, constraints handling, concurrency constraints handling and thread management.

In order to capture the real-time requirements of an object, we consider the following three points: (a) how concurrency constraints can be expressed, (b) how quantitative real-time constraints can be described and (c) how real-time constraints affect the resulting computation model.

Concurrency specification

The concurrency constraints are introduced in the structural model of an application by setting a tagged-value, called concurrency mode, on the operations of classes. The tagged-value is one of the three following: reader, writer or parallel. The concurrency constraint is considered an operation feature and thus has to be specified for each operation of a given class. By default, the concurrency control will consider the worst case.

The concurrency control realized by the local controller is based on the following policy: "1 writer and N readers". This implies that a reading operation may execute in parallel with any other reading operation of the same real-time object, while a writing operation executes in mutual exclusion with any other operations. Parallel operations, involving no concurrency problems, may be executed concurrently with any other operations reading or even writing.

Quantitative real-time specification

The constraints attached to a message may be either one of the following types:

- **Priority Constraint**
This type of constraint represents classical priority values defined as those understood and managed by standard real-time operating systems.
- **Timing Constraint**
This second type of constraint owns different parameters that allow to set: ready-time, deadline, period, period scope or periodicity stop condition of a task and two additional flags allowing customization of the real-time feature attached to a message.

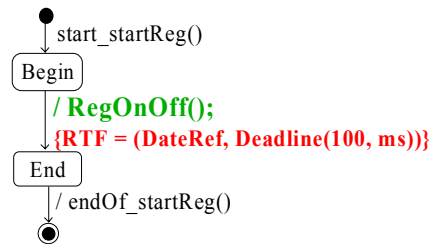


Figure 4 The RTF (Real-Time Feature) tagged value

In order to support this notion, we introduce a tagged value called RTF (Real-Time Feature) Figure 4, which may have values of type priority or timing. This tagged value may be put on elements of type SendAction or CallAction. It can have the following parameters: {RTF=((deadline, ms), (ready-time, ms), (periodicity, ms), NumOfPeriods, endOfPeriod, temporalFault)}. The two last parameters are user functions that may be called respectively at the end of the execution of each period of a periodical task and when a temporal fault is detected.

A specific use of UML statechart for real-time system design

In UML, a statechart owns a context that may be either a classifier or a behavioral feature. Within the AIT-WOODDES approach, statecharts are only used at two levels of granularity to design the behavior of an application (Figure 5):

- Class behavior is described through a restrictive and specialized use of UML statecharts largely based on protocol statecharts as defined in UML semantics.
- Operations behavior is described via an alternate view of statecharts. We do not introduce this view to define our own action language but simply because of the lack of action language in UML.

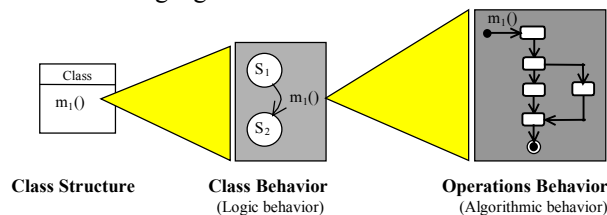


Figure 5 Use of UML State Diagrams at Different Levels of Granularity

Class behavior model: protocol and triggering statecharts

Within the AIT-WOODDES approach, the statechart attached to a class aims at modeling its behavior and can be reckoned under two points of view: protocol view (Figure 6) and triggering view (Figure 7). In UML, objects of an application communicate through message passing that is the result either of a signal raising or of an operation invocation.

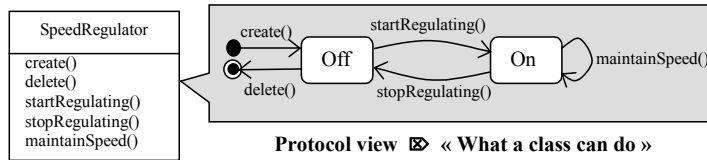


Figure 6 Protocol view of the global behavior

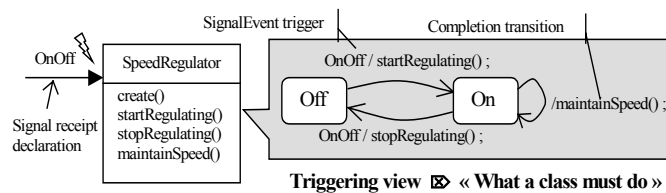


Figure 7 Triggering view of the global behavior

Operation behavior

After having specified the class control logic, the designer models the behavior of the class operations. Up to now, in UML there is no well-defined way to specify algorithms. Since statecharts may be used to specify behavioral features such as methods, i.e. implementation specification of class operations, AIT-WOODDES elaborates the use of statecharts for defining the behavioral specification of object operations.

More specifically, methods are described through a specific use of statechart decomposition in sequences of UML elementary actions: send action, call action, create action, terminate action, destroy action, return action and assignment action.

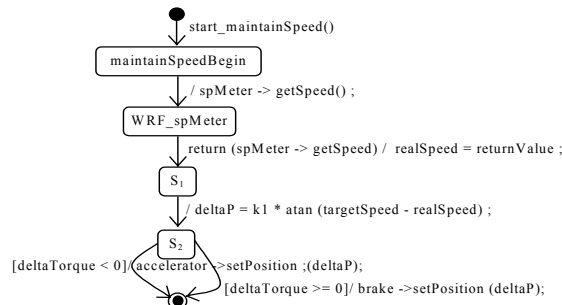


Figure 8 Behaviour of the method implementing the maintainSpeed() operation

Summary

It is a fact that cars and other vehicles are becoming more and more complex with more electronics and embedded applications. Most (if not all) of these systems are critical to the functionality of the vehicle. Especially if we by critical consider the economical issues. Even a sound system in a car that does not function may cost the manufacturer considerable amounts of money. The traditional document/textual based development at the car manufacturers and their subcontractors today is changing into model based processes, as the demands for formalism, tool support and increase of information exchange are necessary. Most of these embedded systems are also distributed over several computing nodes in the vehicle and connected through possibly several networks. Many of them also

have real-time constraints, such as engine control systems, automatic brake and anti-spin systems, etc.

The AIT-WOODDES approach does provide a very attractive solution to the automotive industry by addressing several of the specific issues so important to this domain. We believe that the effort invested in development of automotive electronics will increase and thus be ready to adopt the results from the project.

By integrating the WOODDES methodology into the current development practices it will be possible to achieve a dramatic reduction of the development time and costs of these products. The WOODDES methodology will promote flexibility and reusability through model based development as well as support the capability to produce a large number of variants derived from a generic model. The technical approach adopted in the project is also creating a universal notation adapted to this type of system, which will provide, due to the modelling standards, better facilities for exchange of models and teamwork. It should also contribute to closing the gap between software developers and collaborating teams, by providing a "common language" in their daily work.

References

- [DIPES2000] - "Efficient system modeling of complex real-time industrial networks using the ACCORD/UML methodology", S.G rard, N.Voros, C.Koulamas, F.Terrier. Presented at Architecture and Design of Distributed Embedded Systems (DIPES 2000), Paderborn University, Germany, 2000.
- [ACCORD98] - "Real-Time Modeling with UML: The ACCORD Approach" by A. Lanusse, S. G rard, and F. Terrier, presented at "UML98", Mulhouse, France, 1998.
- [GERARD00] - "Mod lisation UML ex cutable pour les syst mes embarqu s de l'automobile" PhD report by S. G rard, Evry, France, 2000.
- [RTS2000] - "Refinement of UML for Real-Time Modeling with Active Objects" by S. G rard, F. Terrier, and A. Lanusse. Presented at RTS'2000, Paris, 2000.
- [YONE97] - "Object-oriented concurrent programming" by A.Yonezawa, M.Tokoro, Computer Systems Series, MIT Press, 1987
- [SDL93] - ITU-T Recommendation Z.100, "Specification and description language (SDL)", 1993
- [UML] - Introduction to OMG's Unified Modeling Language (UMLTM), 2002-04-23, http://www.omg.org/gettingstarted/what_is_uml.htm
- [WOODDES] - AIT WOODDES Project Homepage, 2002-04-23, <http://wooddes.intranet.gr>
- [IST] - Information Society Technologies Programme, 2002-04-23, <http://www.cordis.lu/ist/home.html>
- [SELIC94] - "Real time Object-oriented Modeling" by B. Selic, G. Gullekson, and P. T. Ward, John Wiley & Sons, Inc., 1994.
- [DOUGLASS99] - "Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns" by B. P. Douglass, Addison Wesley, 1999.
- [ARTHAUD95] - "OMT-RT: Extensions of OMT for better describing dynamic behavior" by R. Arthaud, presented at the 16th International Conference on Technology of Object Oriented Languages and Systems (TOOLS EUROPE'95), Versailles, France, 1995.
- [AWAD96] - "Object-Oriented Technology for Real-Time Systems: A Practical Approach Using OMT and Fusion" by M. Awad, J. Kuusela, and J. Ziegler, Upper Saddle River, NJ 07458, USA: Prentice Hall, 1996.
- [MOORE00] - "Why tasks aren't objects and how to integrate them in your design" by A. Moore, Embedded Systems, pp. p18-30, 2000.
- [TERRIER96] - "A Real Time Object Model" by F. Terrier, G. Fouquier, D. Bras, L. Rioux, P. Vanuxeem, and A. Lanusse, presented at TOOLS Europe'96, Paris, France, 1996.