Product Lifecycle Management and System Engineering Airplane Electrical Power Management

Olivier Forner

Session 6A: Architectures

ERTS 2004

Nom de l'intervenant : .... Forner ....
Prénom : ....Olivier………………………………………….….……
Fonction : ....Consultant Senior..................................…......
Adresse : 22 Quai Galliéni..............…...........…........
Code postal : 92158 SURESNES Cedex, France........…...…....
Société : ....Dassault Data Services……………...…….
Pays : ……………………………………………………………………………………………...…......
Tél. :……+33 6 63 37 36 10........…................ **Fax :** +33 1 41 38 41 31
E.mail : o-forner@dassault-data-services.fr **Web :** www.Dassault-data-services.fr......

Biography:

Olivier Forner is a senior consulting engineer at Dassault Data services in the area of System Engineering, especially in Embedded Software methods and tools, and distributed architecture.

He has practical industry knowledge of this domain through work with leading worldwide aerospace, automotive, and telecom companies.

Prior to joining Dassault Data Services, Olivier worked for seven years as consultant and R&D engineer for Object Oriented databases and CAD/CAM software editors and consulting organizations. He was involved in large number of customers projects where he had the opportunity to address key challenges related to PDM/PLM implementations.

- Please find enclosed a white paper of less than 10 pages
- The title of the paper is :

…………………………………………………………………………………………………….....…....

Product Lifecycle Management and System Engineering

Airplane Electrical Power Management

…………………………………………………………………………………………………….....…....

Abstract

Industrial are more and more faces to the hard problem of developing more sophisticated real-time systems while time to market, quality and cost constraints are constantly increasing.

The **product development challenges** they have to meet with are:

- Accelerating rate of innovation and technology changes
- Increasing pressure for delivering products faster
  - 80% of the market share generally goes to the first two companies to market a new product
  - Do it right the first time
  - Control risk associated to fast delivery
  - Drive components platform/technology reuse
- Increasing product & process complexity
  - Increasing number of requirements coming from multiple sources
  - Increasing regulation complexity
  - Increasing interactions between systems
  - Increasing software functionalities
  - Increasing product variants
  - Increasing multi-disciple interaction (Mechanical / Electronic/Software)
  - Unpredictable fluctuations
- Increasing pressure on product cost reduction
- Increasing pressure on higher quality

Complex real-time systems are characterized by :

- Extreme dependability (reliability, availability)
- Predictability
  - For example, critical systems must be deterministic, for safety reasons.
- Diverse and feature-rich functionality
- Continuous feature upgrades (evolutionary requirements)

- Physical distribution

This complexity requires focused support in at least the following areas :

- Timeliness and performance
- Time-aware communication
- Concurrency management
- Resource optimization
- Distributed system
- Fault tolerant (detection, treatment, analysis, recovery)
- Architectural design
- System certification

This paper will outline and discuss some of these trends through the presentation of an aeronautical project.

We have been involved in a preliminary technical development, three years ago, which deals with the following requirements:

- Airplane Electrical Power Management
- Provide guidelines for the production of software for airborne systems and equipment.
    - Assess new development method and CASE tool in order to improve productivity.
- Dependability
- Comply with airworthiness requirements define within the DO- 178B standard.
- Assess new technological concepts and new architectures in order to improve the competitiveness of future airplanes and develop the prototype
    - Integrated Modular Avionics

Airplane Electrical Power Management

The airplane electrical power management consists of the following applications:

- Load management Unit,
- AC/DC commutation
- Protection
- Maintenance

The goal is mainly to prevent damage and fires from electrical and to distribute the power to all the technical and commercial facilities.

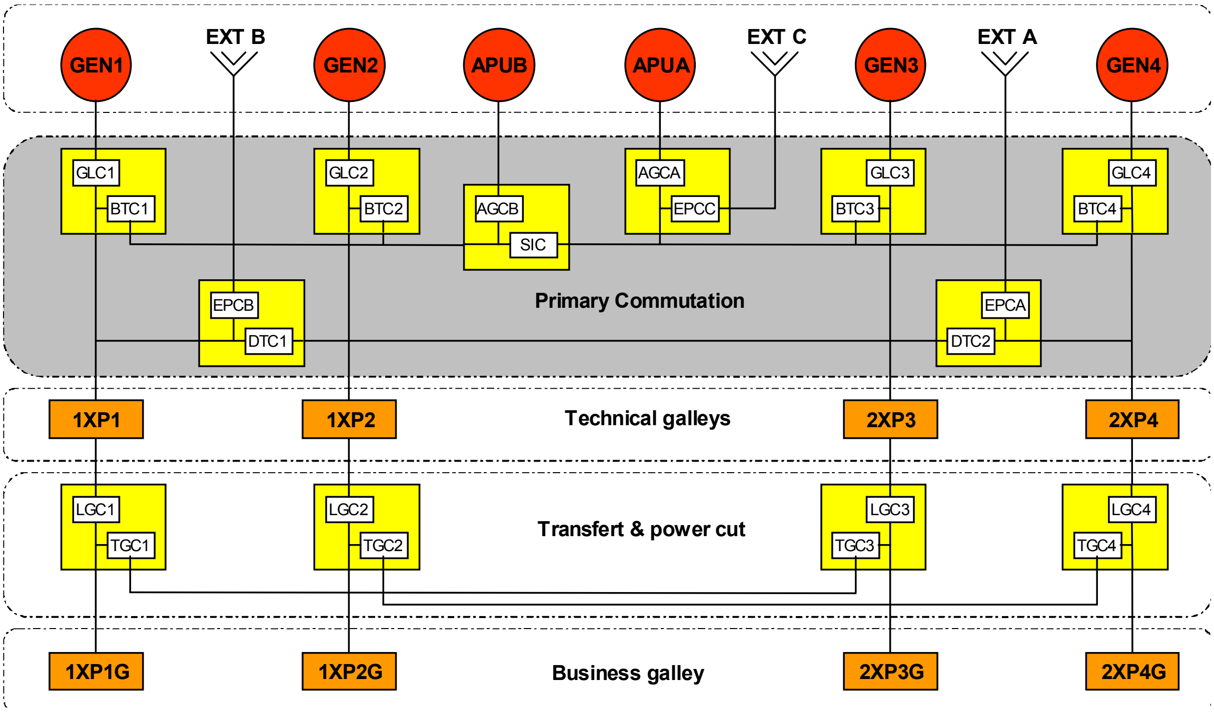

Figure 1: AC network architecture

Aircraft manufacturer and Airline Company need to have more flexibility to define and set up the facilities configuration and the electrical master box.

- The electrical master box is composed of contactors and protections.
- The facilities configuration is loaded from the ground using the ad hoc operator interface.

This flexibility enables an aircraft manufacturer to "reuse" the electrical system.

New development method and CASE tool

What about using an object oriented methodology and defining a dedicated case tool for developing these software applications?

The main objectives are the following:

- Conciliate new development method and tools with the DO-178B standard,
- Provide a shared vision of the system within the consortium,
- Take into account new requirements during the first two third of the project.

In order to achieve them, we have performed the following tasks:

- First, all the DO-178B software plans have been developed
- The software life cycle, including the methods and tools to be used for the activities of each software life cycle process have been selected and set up

summary previous next

- The AGLAE workshop has been defined (see figure below)
- The added value was the optimal integration of all the COTS

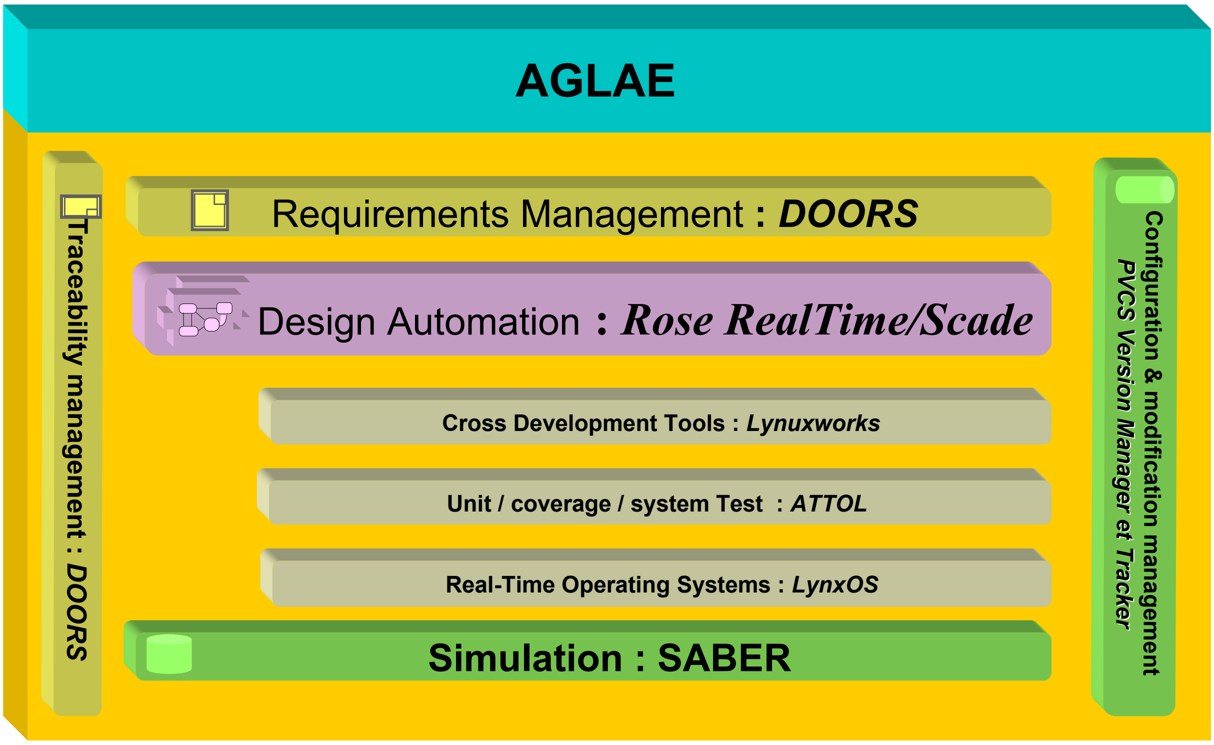

Figure 2 Hard Real Time Case Tool

The Object Oriented Development has been done using the AGLAE workshop:

- Analysis /conception using UML profile real time
- Target C++/ C
- Code / test / documentation generation
- Cross Development NT/LynxOS

The commutation application is usually carried out with logic wired technology. We intend to implement it with software.
To facilitate and shared the system behavior before its development, a virtual test bench has been set up to validate the specification.

- The electrical behavior is simulated with Saber,
- The electrical power management is performed from the control unit

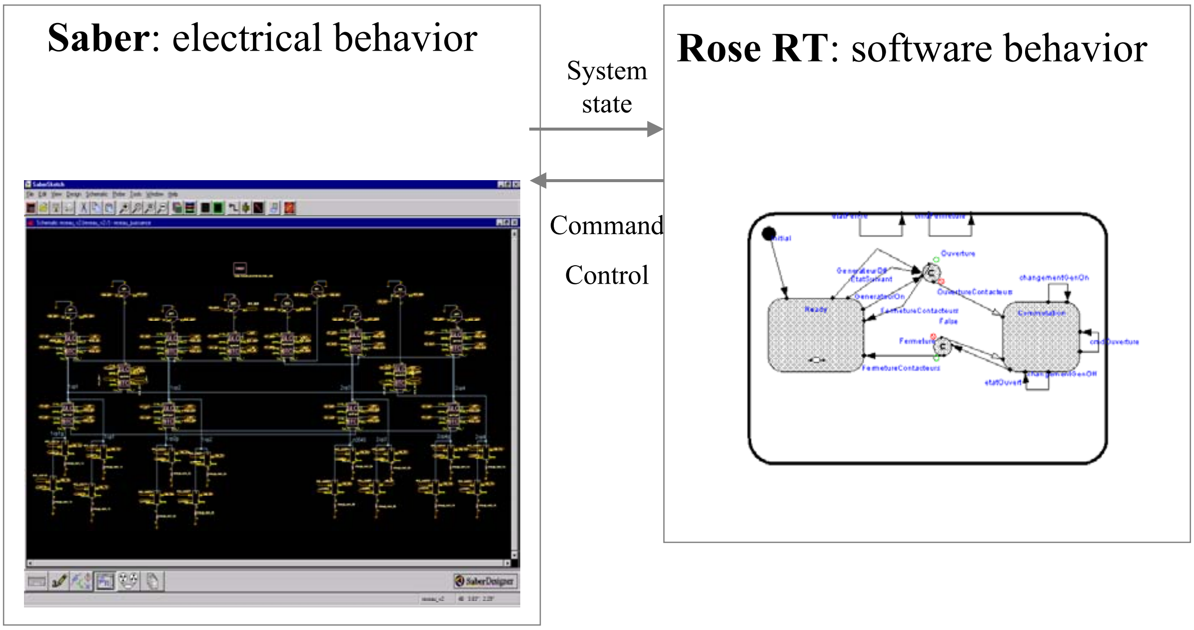

Figure 3: Specification validation using a virtual test bench approach

Dependability

The dependability of the system is done through Control Unit redundancy. The software development must handle multi control units requirements.

Model Driven Architecture

The OMG has formulated a vision of a method of software development based on the used of models.
Key characteristics of MDA:

- The focus & principal products of software development are models rather than programs
- "The design is the implementation" (i.e. UML as both a modeling & implementing language)

UML plays a critical role in MDA

- Automatic code generation from UML models
- Executable UML models

We will explain in our oral presentation how we have achieved the dynamic load management sub system development using all these different models

summary previous next

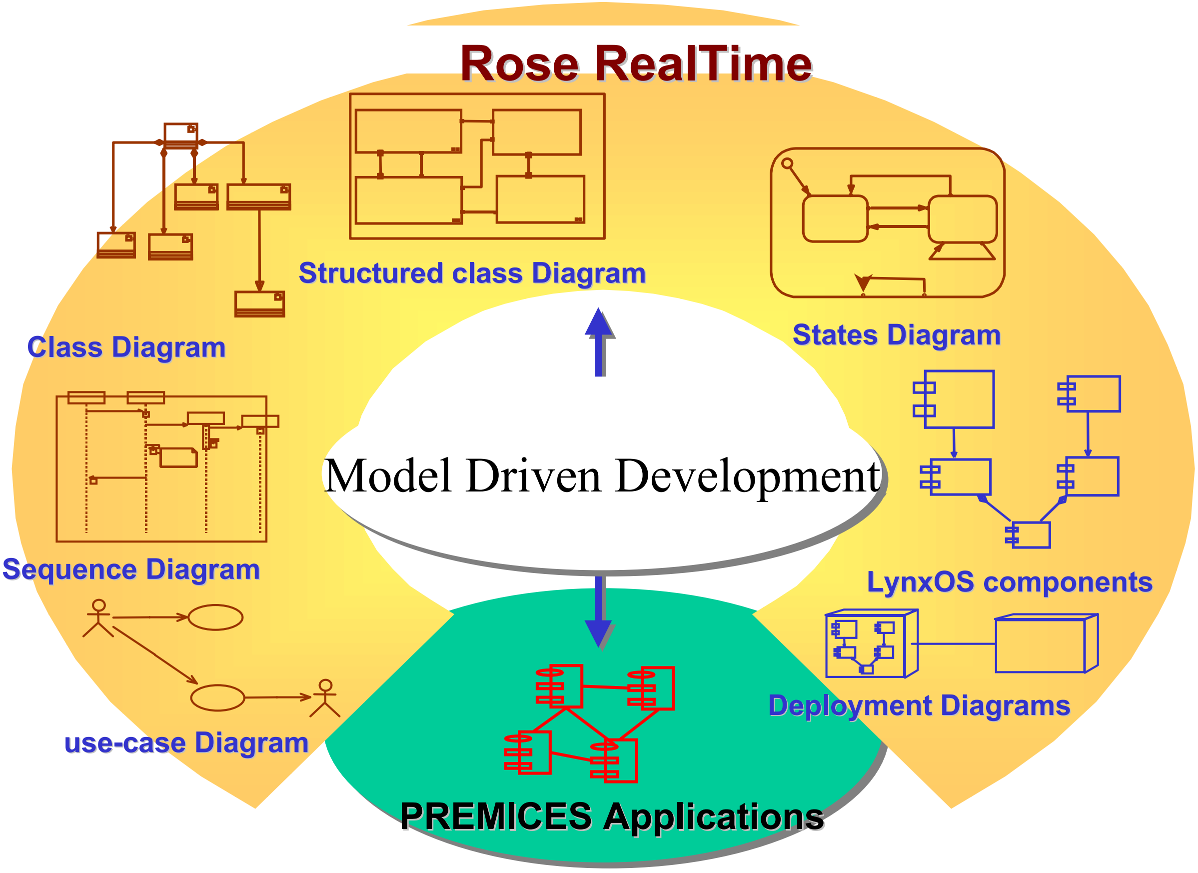

Figure 4: Model Driven Development

Architecture description language

Good architectures are essential for the successful construction and evolution of complex real-time systems.
Architectural designs must be visible and enforceable or "architectural decay" sets in.
The concepts of active objects, ports, connectors, and protocols provide powerful modeling capabilities for capturing our architectures. Furthermore, automatic enforcement can be achieved through automatic code generation from such models

summary previous next

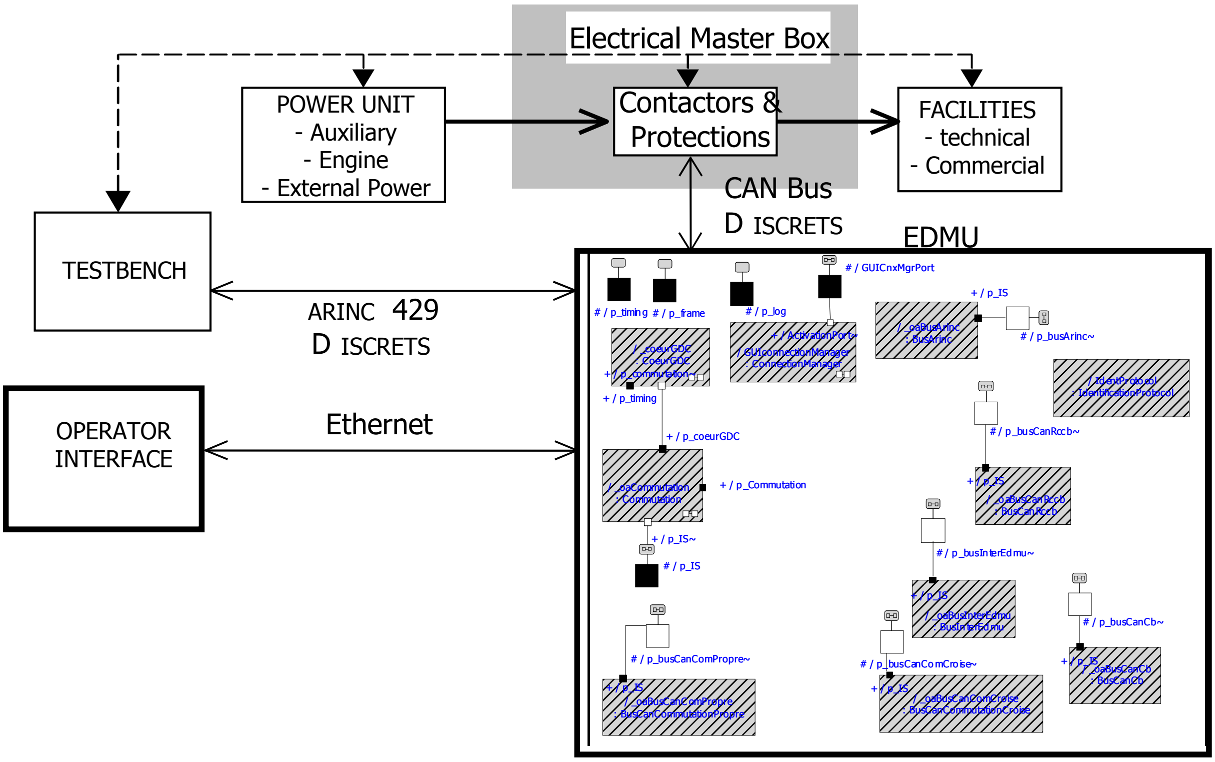

Figure 5: Architectural Language Description

Using Executable Models

Complex real-time systems can be very difficult to understand and to design. The technical process involved should take into account human cognitive processes

The use of automation (executable models and automatic code / test / documentation generation) has significantly reduce the development cycle, especially the time required to reach critical knowledge threshold and the time required to convert a design into an implementation.

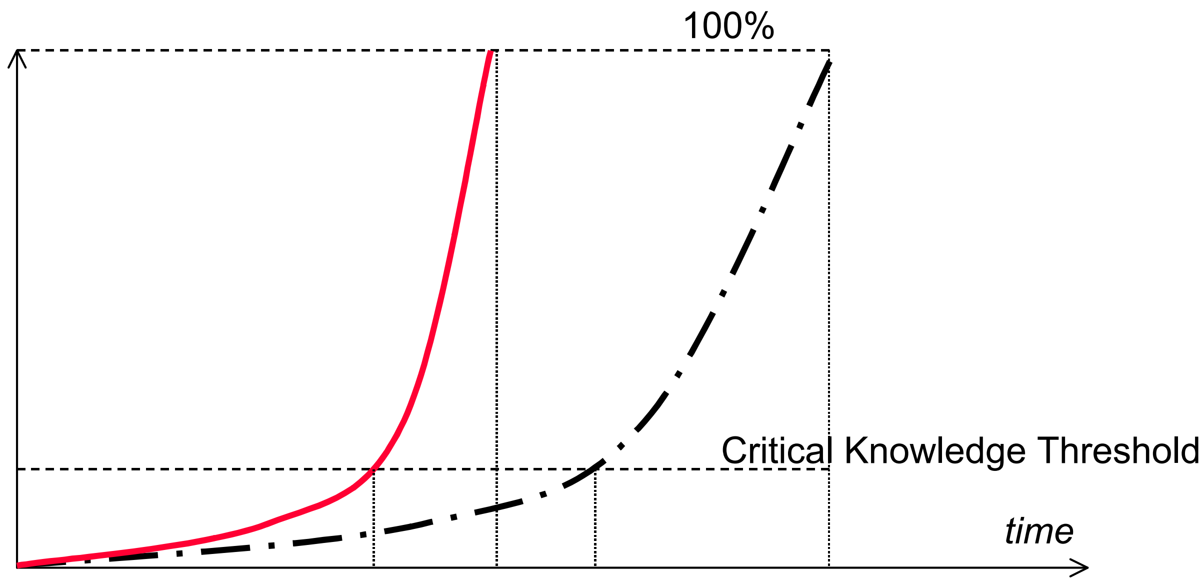

Figure 6: Reduced Time development

The development process was iterative and incremental, mainly due to the fact that we have to take into account new requirements during the first two third of the project.
The figure below illustrates the added values of using models and code / test / documentation generation.

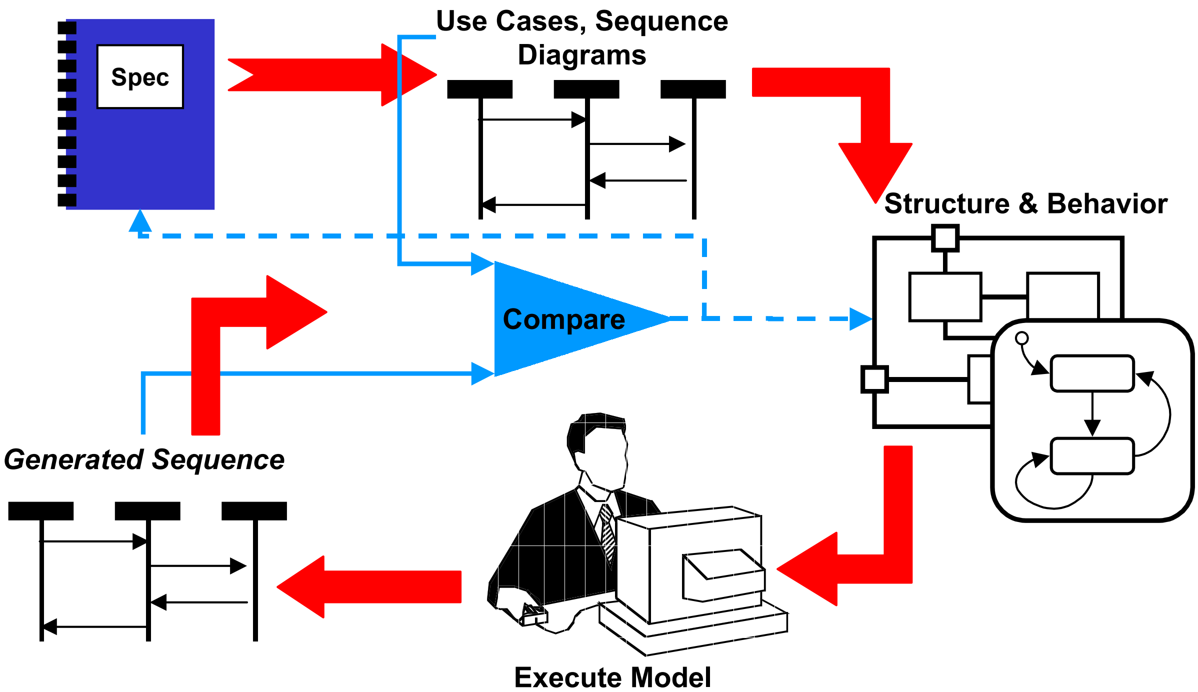

Figure 7: Iterative/ incremental models

Performance Tuning

Yes, but what about performance and performance tuning?

- What is the cost of abstraction ?

How about modeling real-time specific phenomena?

- Time and timing mechanisms
- Resources (processors, networks, semaphores, etc.)

Exploiting current real-time system theory?

- Schedulability analysis (e.g., rate-monotnic theory)
- Performance analysis (queueing theory)

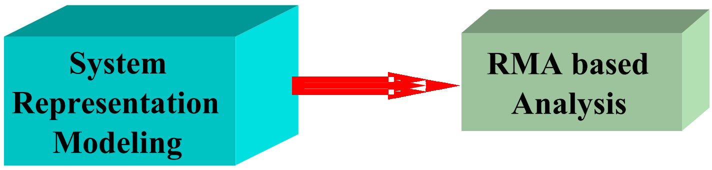

Figure 8: Rate Monotonic Analysis

First of all, Rate Monotonic Analysis is the applicable technology used within the project to predict timing performance.

Secondly, our scheduling policy is divided in two levels :

- At the LynxOS level which manages processes and physical threads (priority …)
- At the thread level with a dedicated scheduler (in order to support Run Time Completion paradigm).

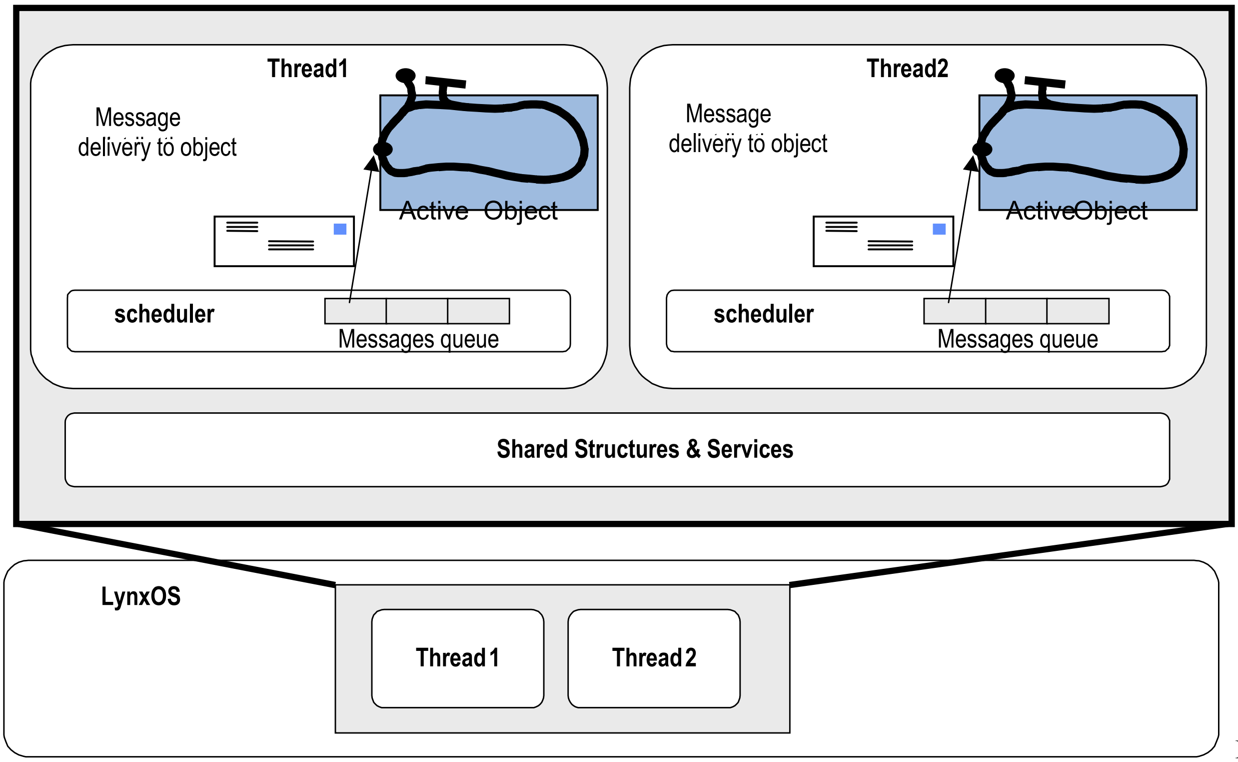

Figure 9: Intra thread scheduling

The performance-tuning phase consists mainly in:

- Tuning algorithms (complexity analysis)
- Providing optimized mechanisms for inter-process and network communications
- Designing allocation of active objects within logical threads
- Mapping logical threads to physical threads and processes
- Reducing memory footprint (Object layout …)
- ….

Wrap up

The lessons learned during this project are:

- Use of UML Real Time for Hard Real Time system,
- High responsiveness to requirements change,
- Tools Integration within the AGLAE Workshop,
- Software productivity,