THALES

Experimentation of MARTE in the Industry

An early adopter case study

Sébastien Demathieu
Thales Research & Technology
[sebastien.demathieu@thalesgroup.com]

MARTE fosters model-based design of real-time and embedded systems in Thales

- Why we need a UML profile for Real-Time and Embedded
- Involvement of Thales in the definition of MARTE
- Case study: a generic Flight Management System
- Applying MARTE to design a generic FMS

UML MARTE

THALES

Why we need a UML profile for RTE

UML is emerging as a possible solution to address the Real-Time and Embedded domain

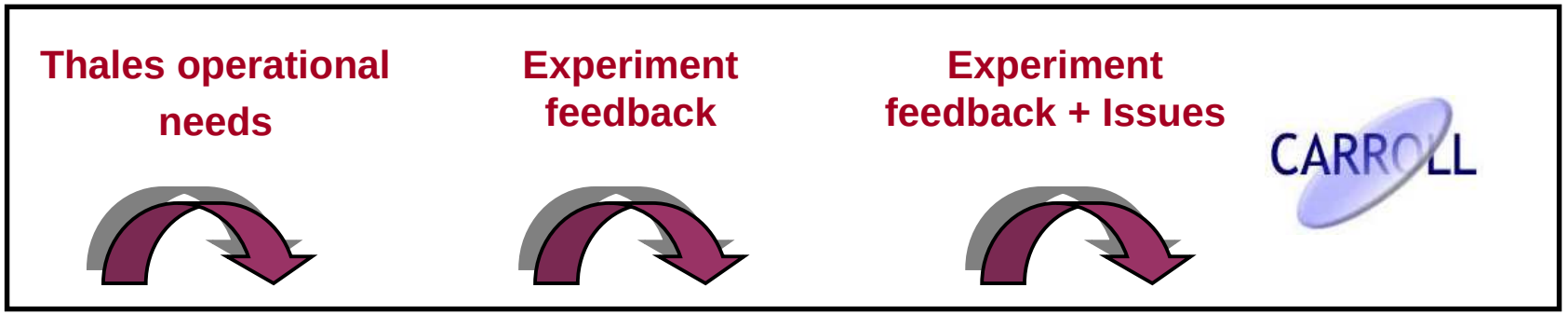
- A large audience in the Software Engineering community
- Extension capabilities through UML profiles (e.g. SysML)
- But lacks key notions to fully address RTE specifics (time, resource, scheduling)

Previous attempts to adapt UML to the RTE domain

- Academic initiatives
- Commercial tools: ARTiSAN, Rational Rose RT, Rhapsody (Real-Time UML)
- UML profile for Scheduling, Performance and Time (SPT)
  - The first OMG adopted specification in this domain
  - Defines annotation mechanisms to perform quantitative analysis
  - Required major improvements over time

Extensions to UML 2 are required to deal with modeling and analysis of Real-Time and Embedded systems

Involvement of Thales in MARTE



Request For Proposal definition — 02 / 2005

Initial Submission — 10 / 2005

Revised submission adopted by TC — 06 / 2007

Finalization Task Force — Today

First experiments on software design

SYSTEM@TIC PARIS-REGION Pôle de compétitivité

Experiments on platform modeling + Scheduling analysis

Next steps:

Deployment of MARTE

Application to Systems Engineering

THALES

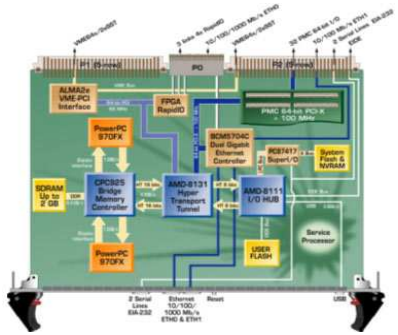
The case study: a generic FMS (1)

A software-intensive avionics system

- Automated navigation (guidance for related devices: AP / FD / ATH)
- Lateral and vertical trajectory optimizations
- Fuel consumption and forecast
- Flight data display

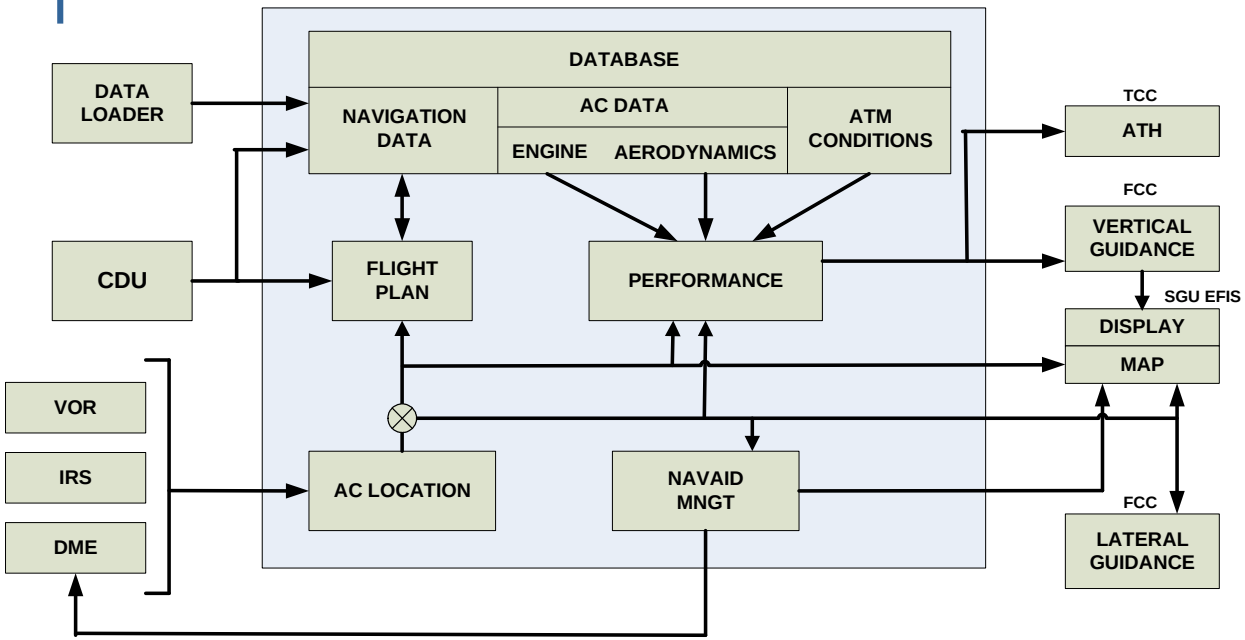
A safety-critical, hard-real time, embedded system

- Strict latency on processings and communications
- Strict jitter constraints
- Dimensioning of buses, FIFOs and CPU power

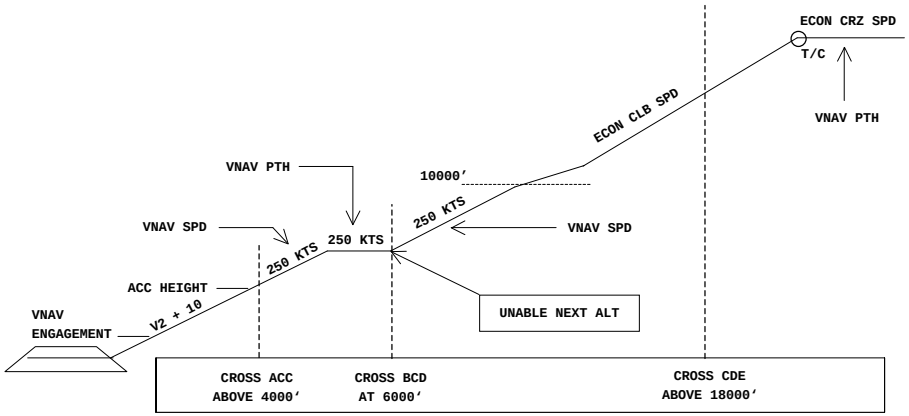


THALES

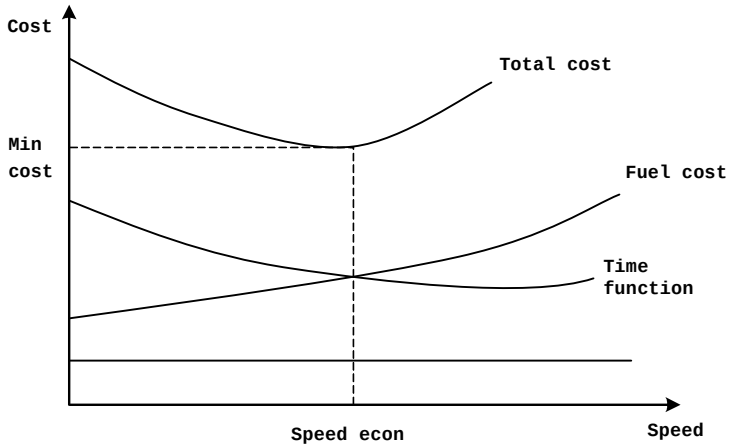
The case study: a generic FMS (2)



FMS block diagram



Lateral and vertical navigation



Fuel consumption / cost index

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Tool support

Early experiments of MARTE required to develop our own tooling

- MARTE profile implemented in various UML tools (Rhapsody, IBM RSA, MagicDraw)
- Value Specification Language (VSL) editor consolidated from Papyrus
- RapidRMA gateway developed for model-based scheduling analysis

Tools integrated through Eclipse / EMF / UML2

Implementation of MARTE for RSA available in open source (EPL)

- OMG MARTE web site: http://www.omgmarte.org

Experimentation highlights

System context definition

- Use case overview
- Time constraints on performance scenarios
- Types and units

System design

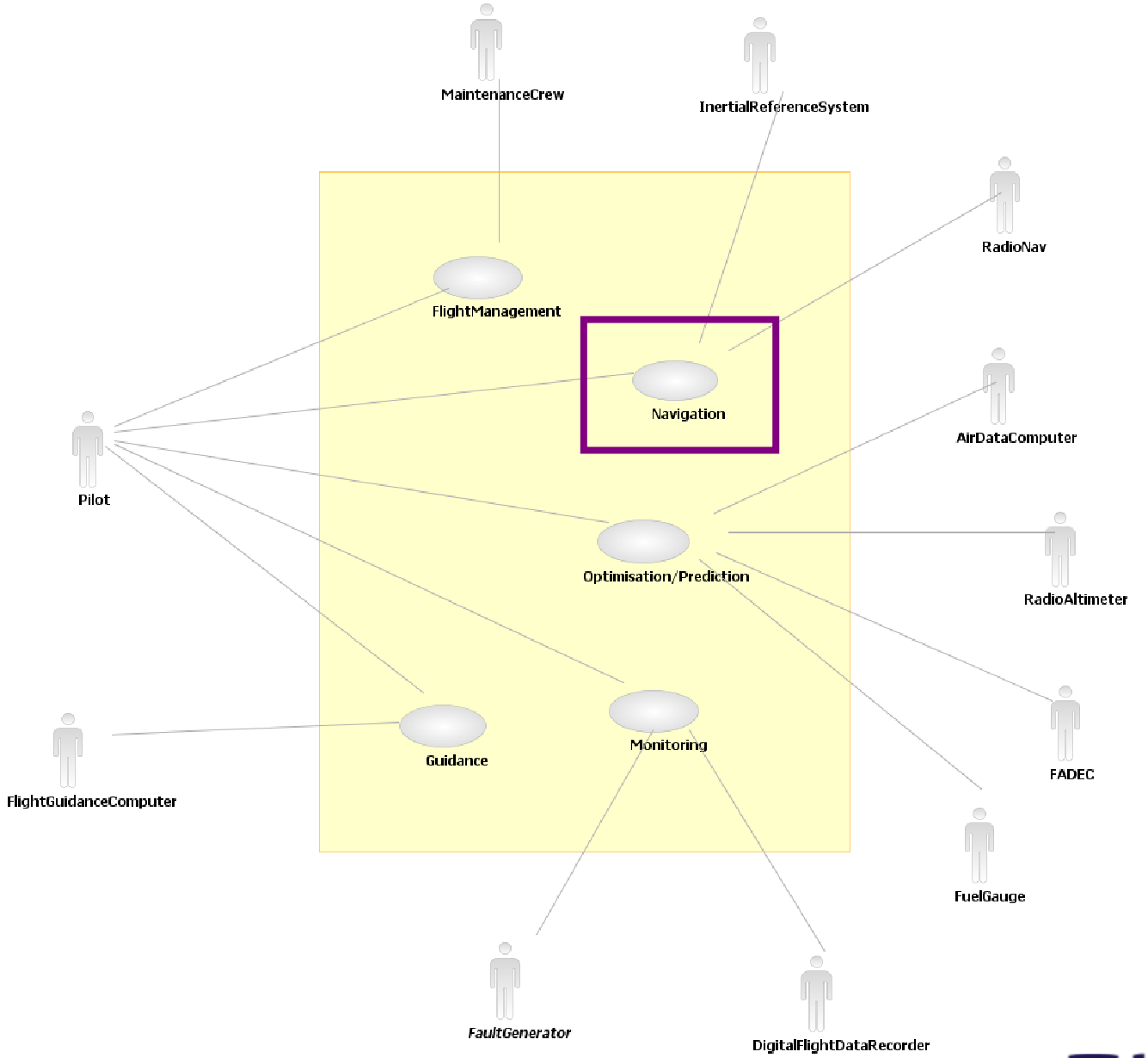
- Logical decomposition (structure, behavior)
- Physical decomposition (resources)
- Allocation

Scheduling analysis

- Building scheduling analysis model
- Validate system scheduling with RapidRMA

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

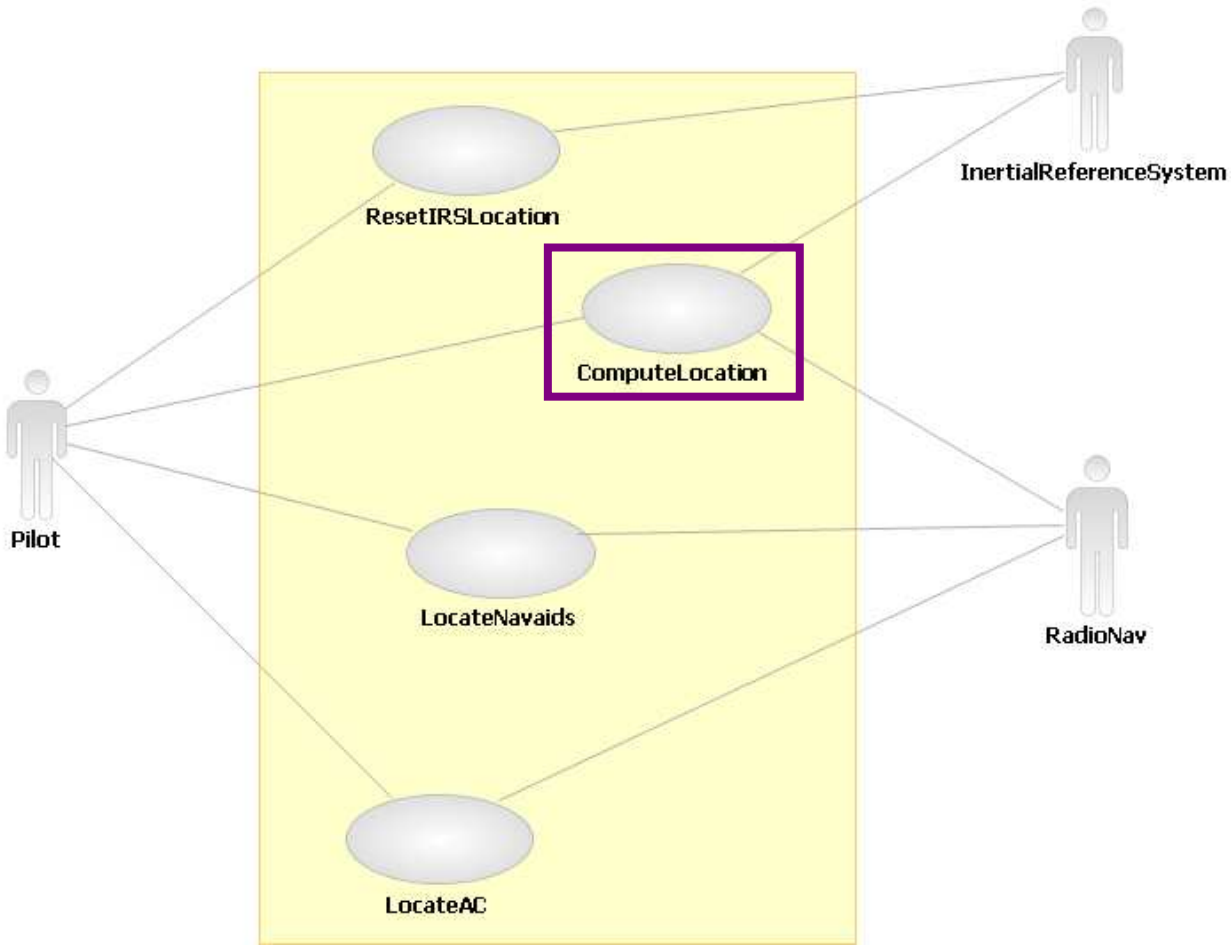
Context and use cases



Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Context and use cases

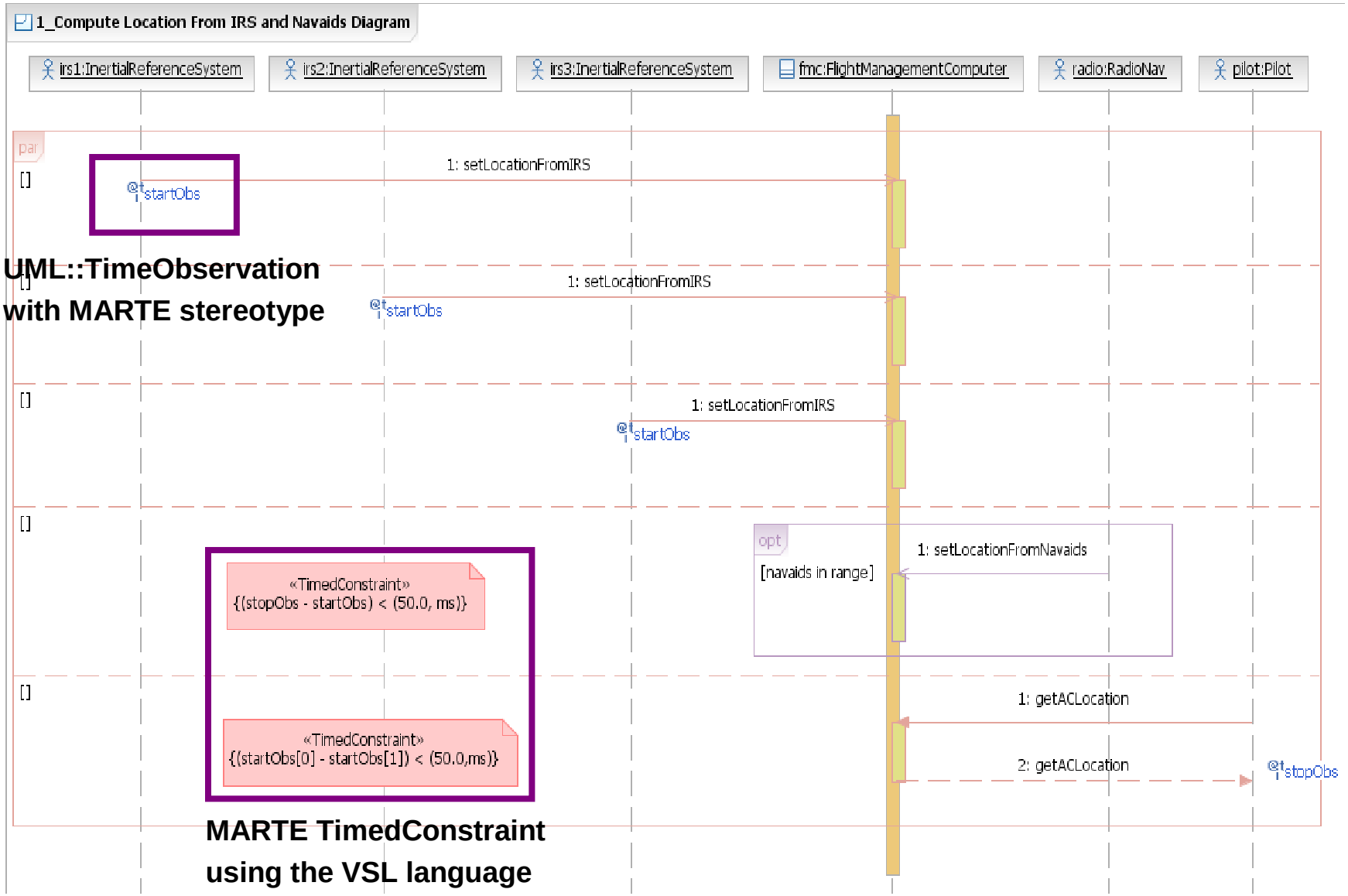


Navigation use cases

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

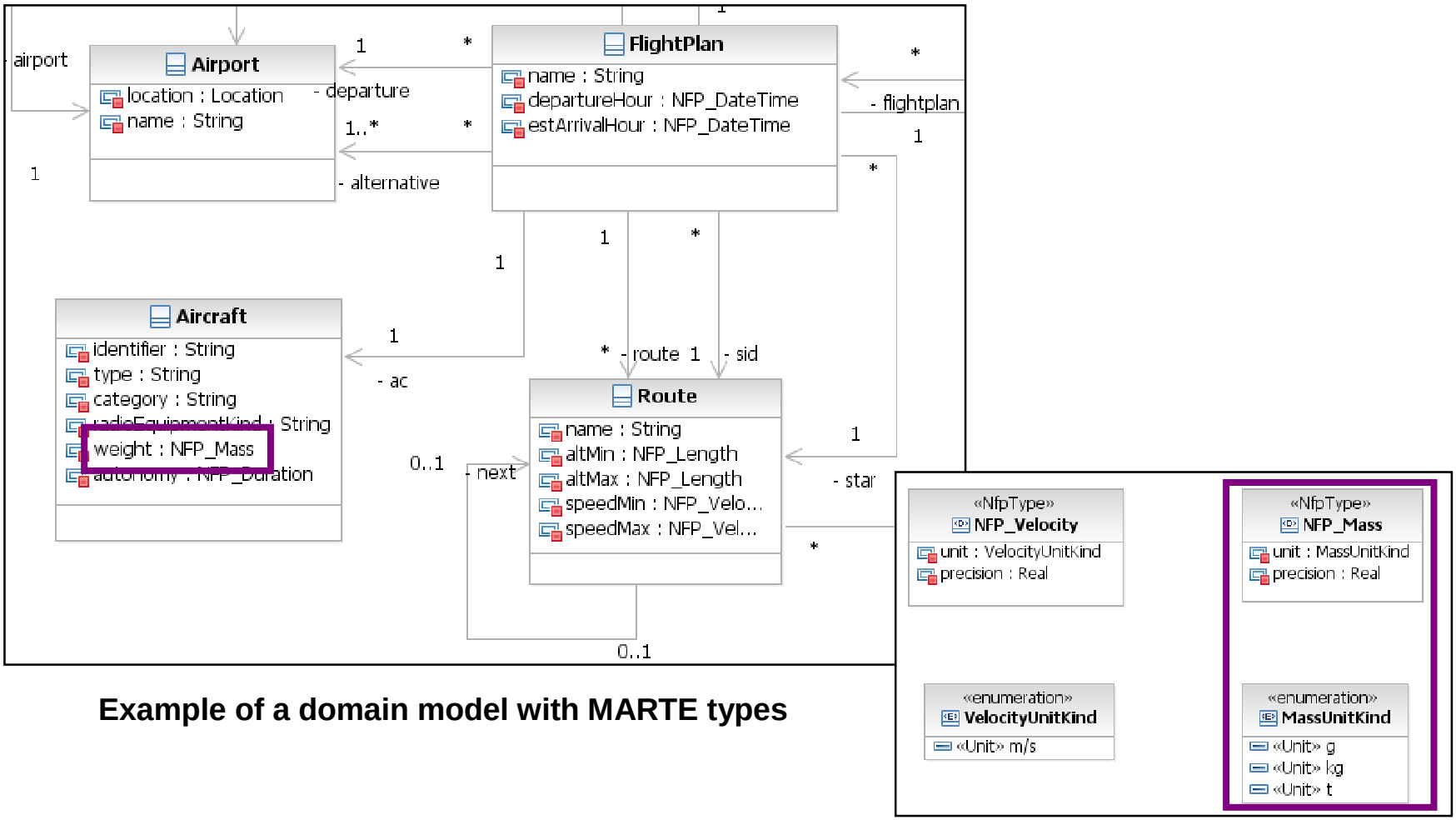
Time constraints on performance scenarios



Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Types and units



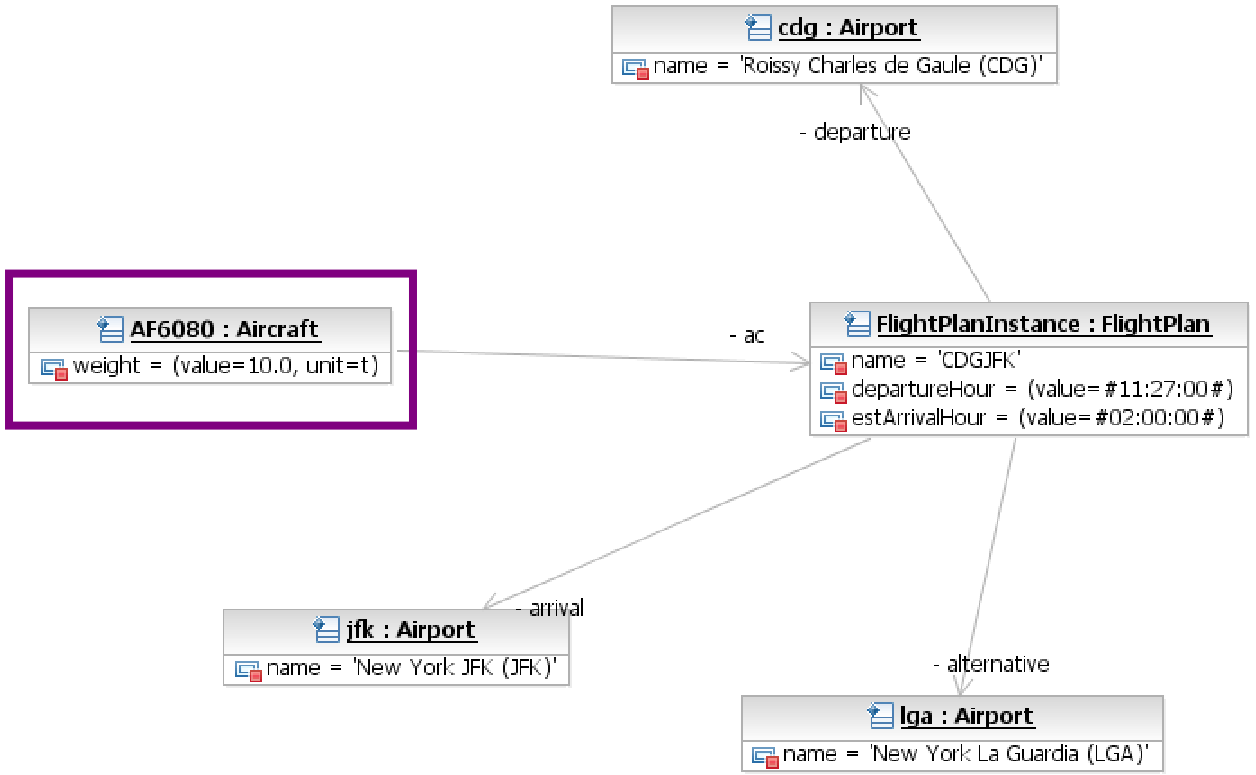
Example of a domain model with MARTE types

User-defined types and units that extend the MARTE Library

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Using types and units in VSL expressions



Units and types used along with value specifications

Experimentation highlights

System context definition

- Use case overview
- Time constraints on performance scenarios
- Types and units

System design

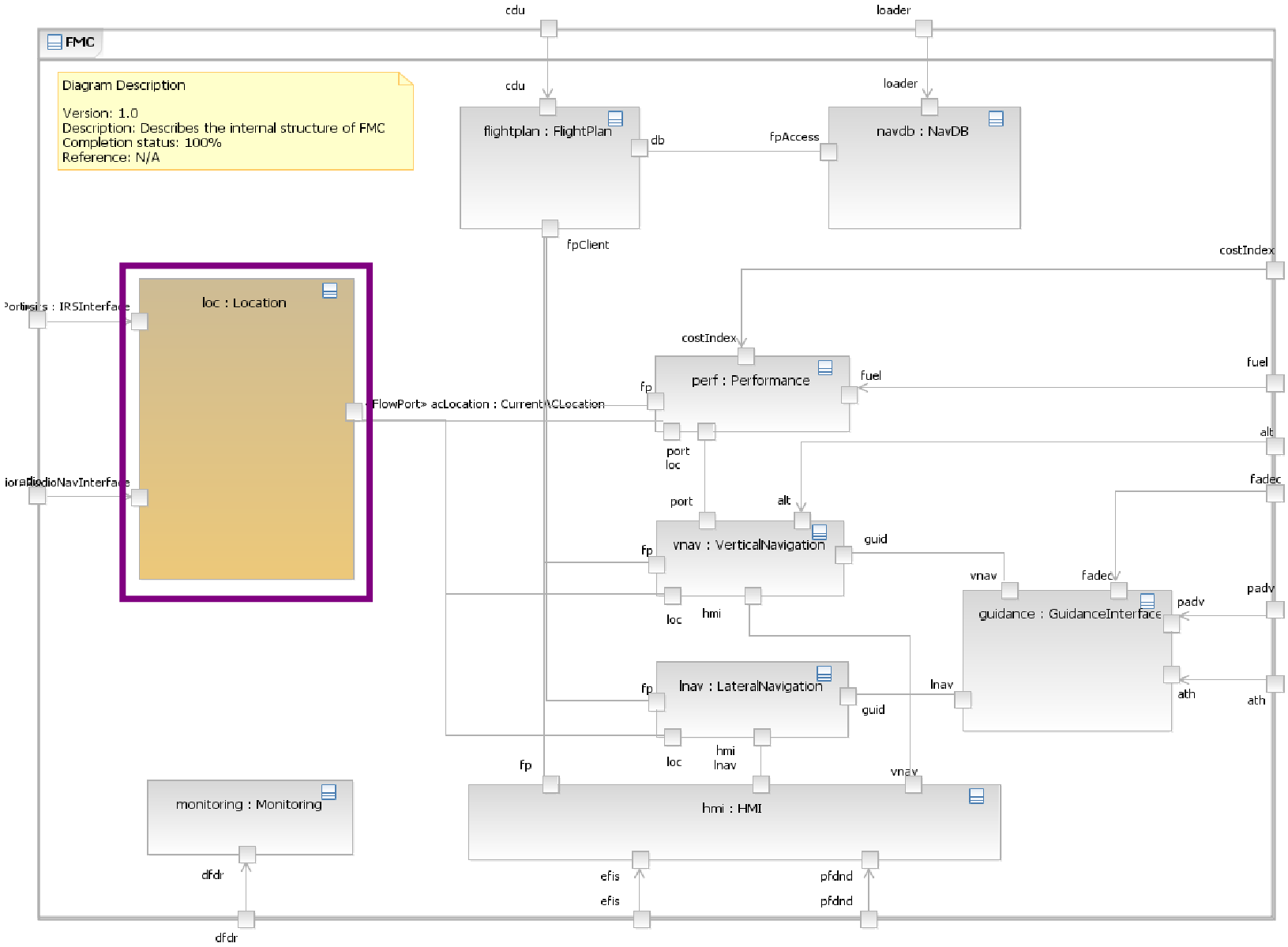
- Logical decomposition (structure, behavior)
- Physical decomposition (resources)
- Allocation

Scheduling analysis

- Building scheduling analysis model
- Validate system scheduling with RapidRMA

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

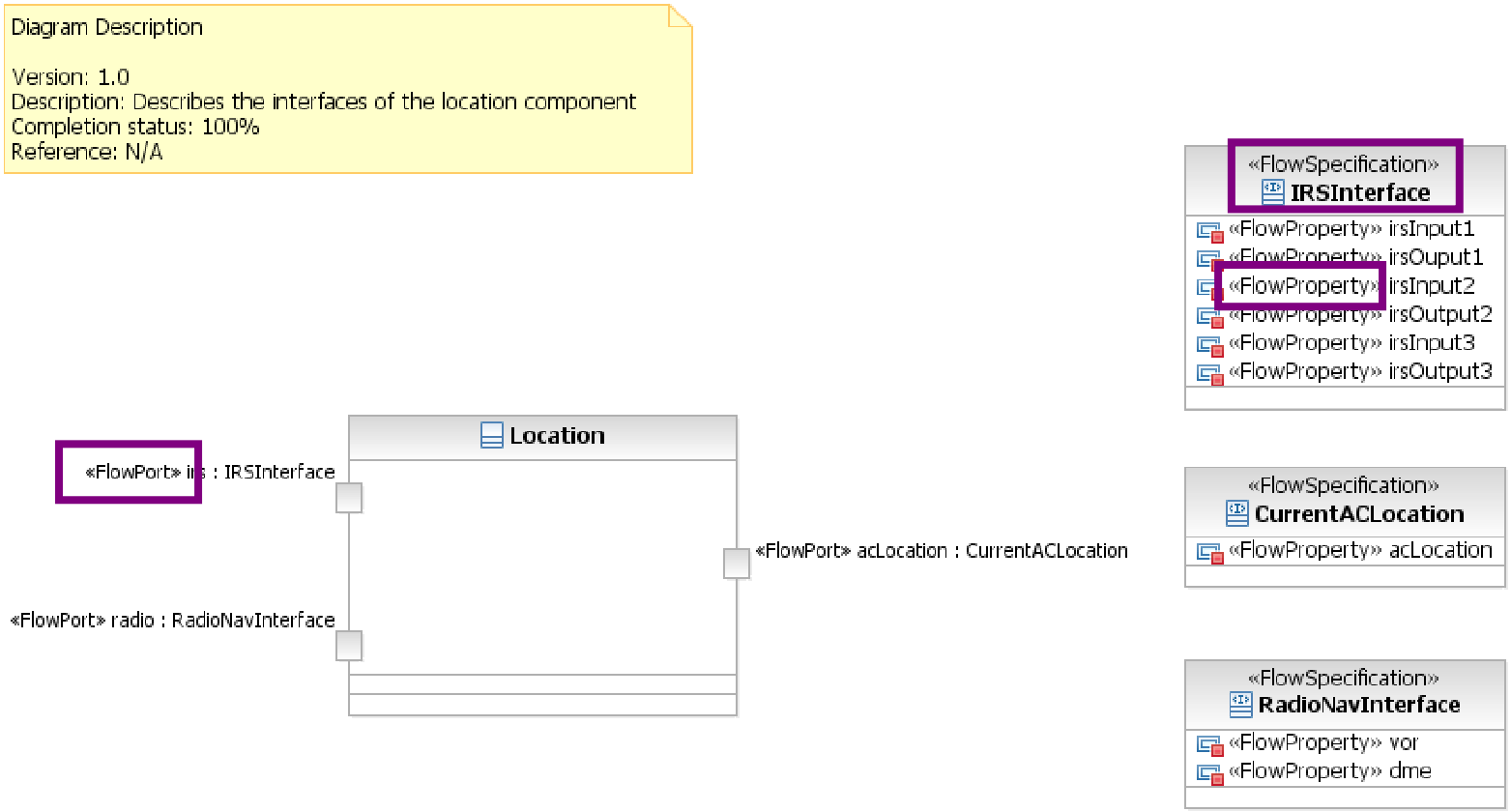
Logical decomposition (structure)



Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Logical decomposition (structure)

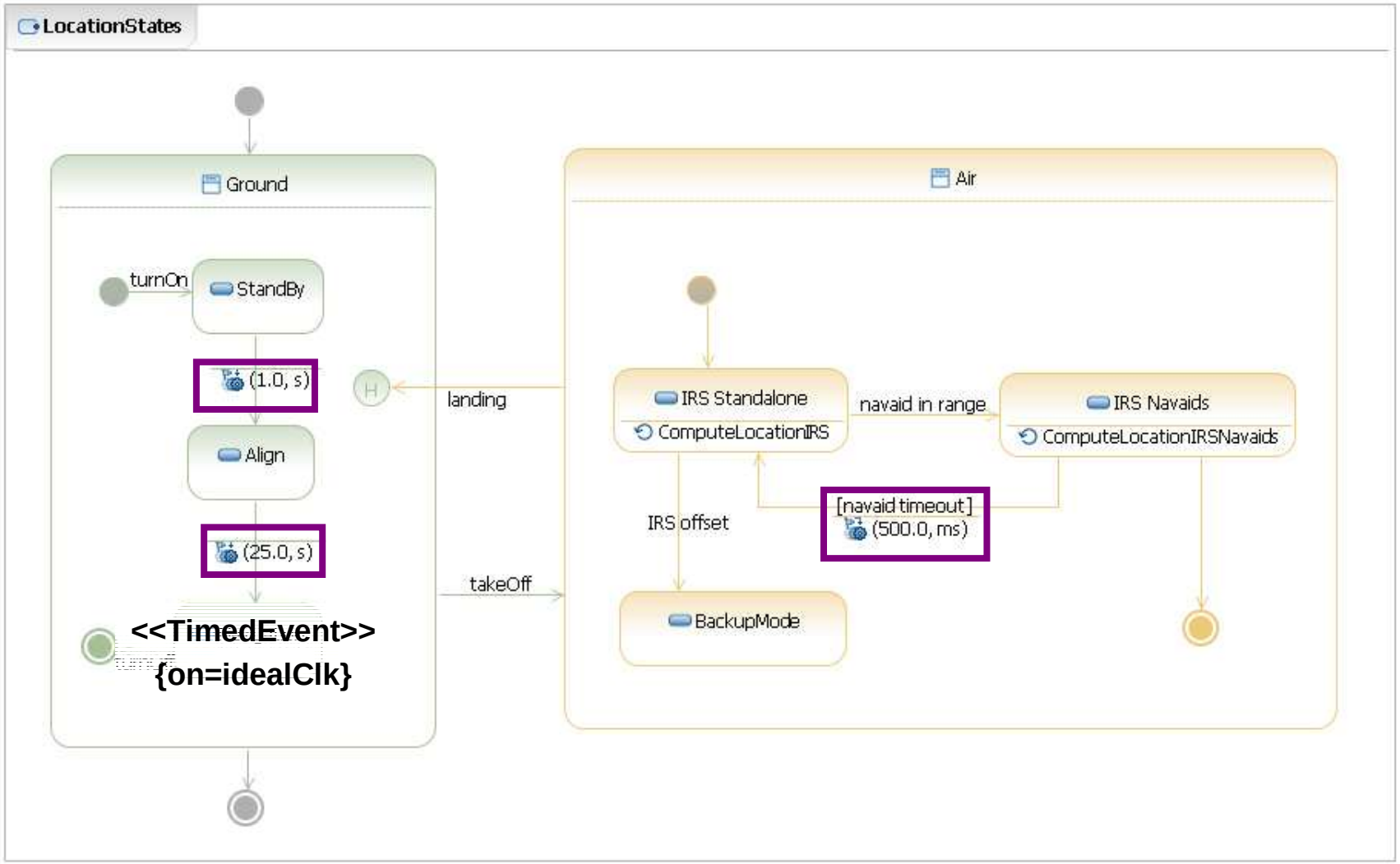


Flow-related concepts inspired from SysML

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Logical decomposition (behavior)

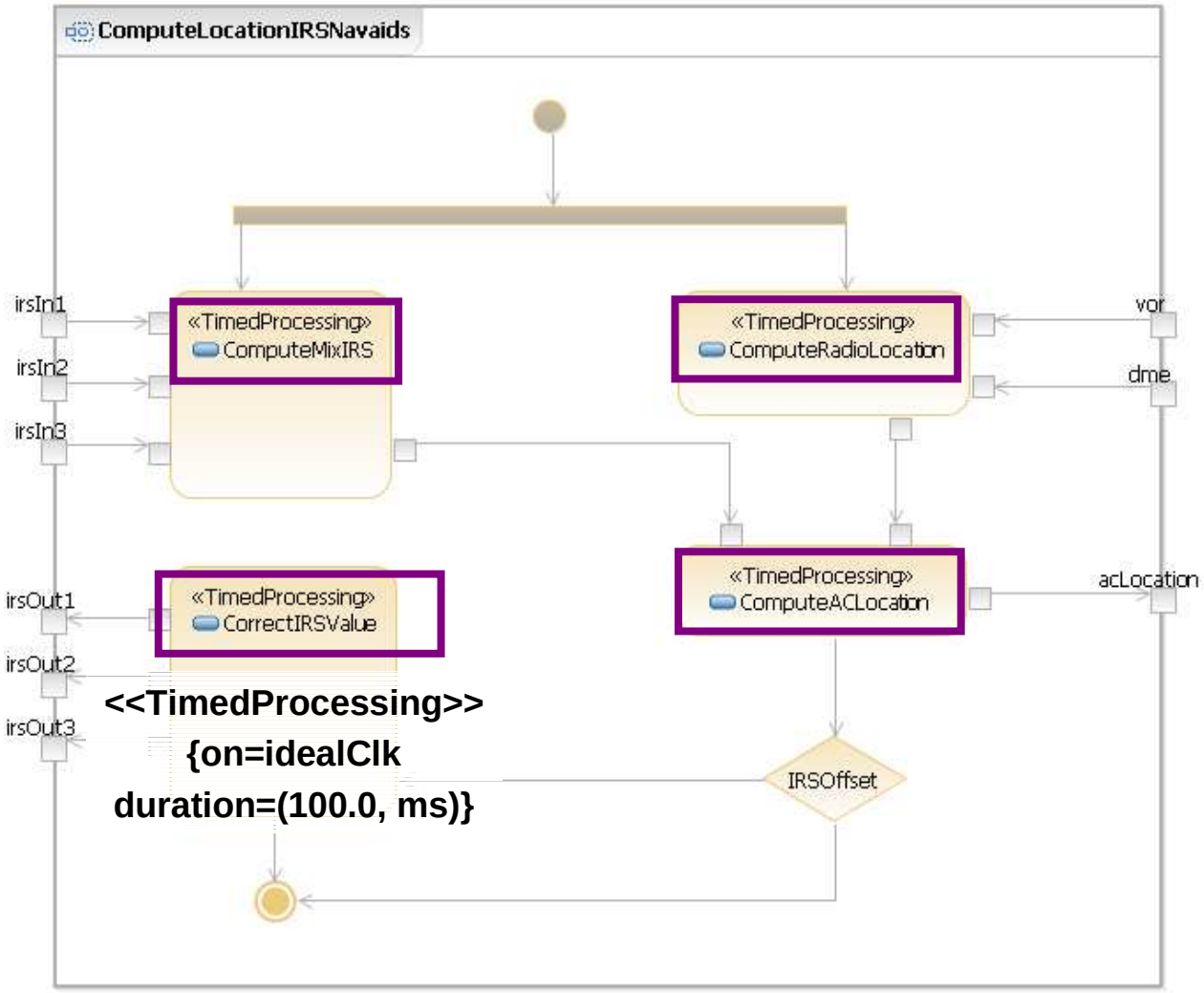


Time-triggered transitions (relies on physical time in this example)

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Logical decomposition (behavior)

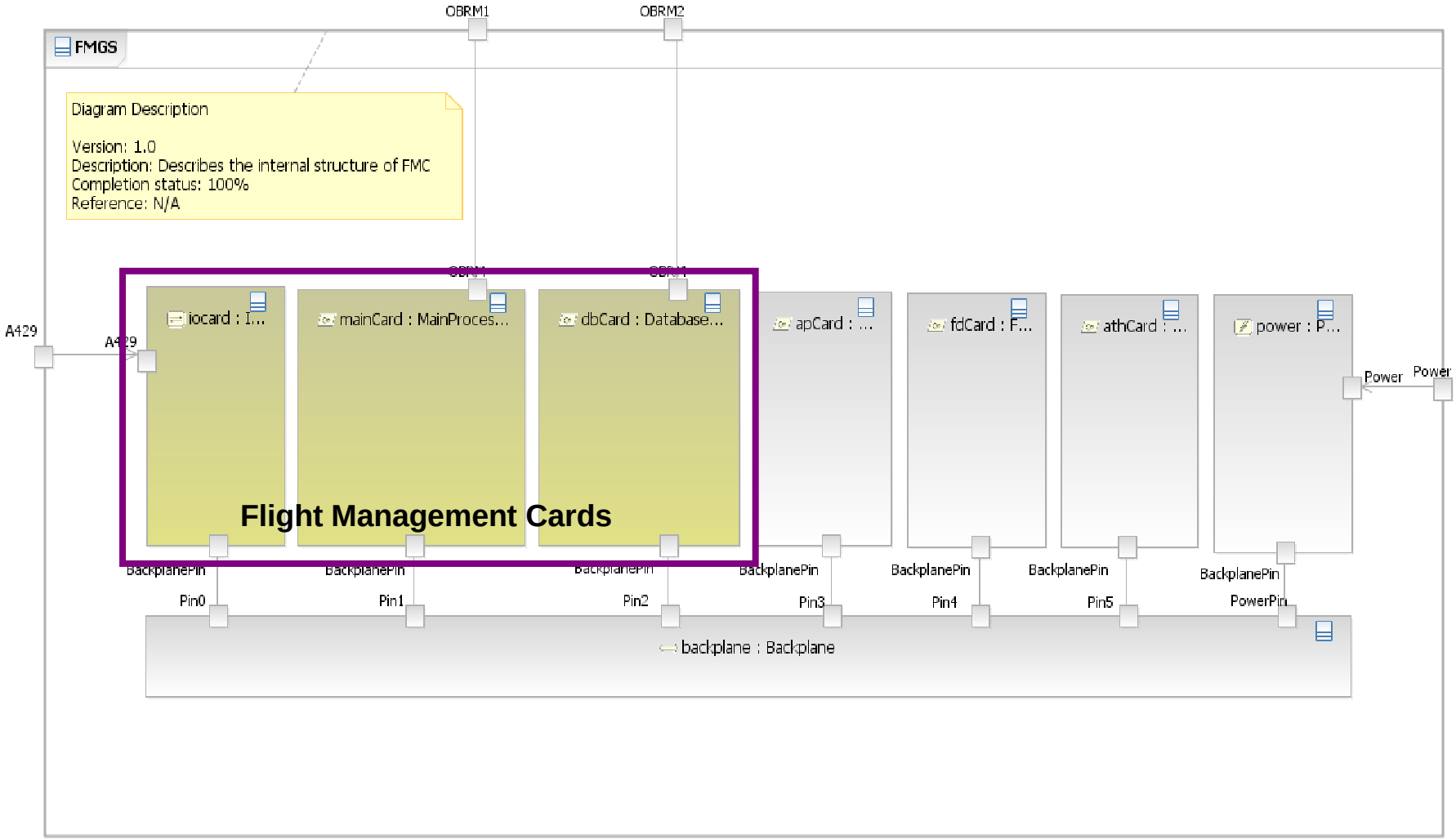


Timed processings (with a duration) in an activity diagram

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Physical decomposition (hardware)

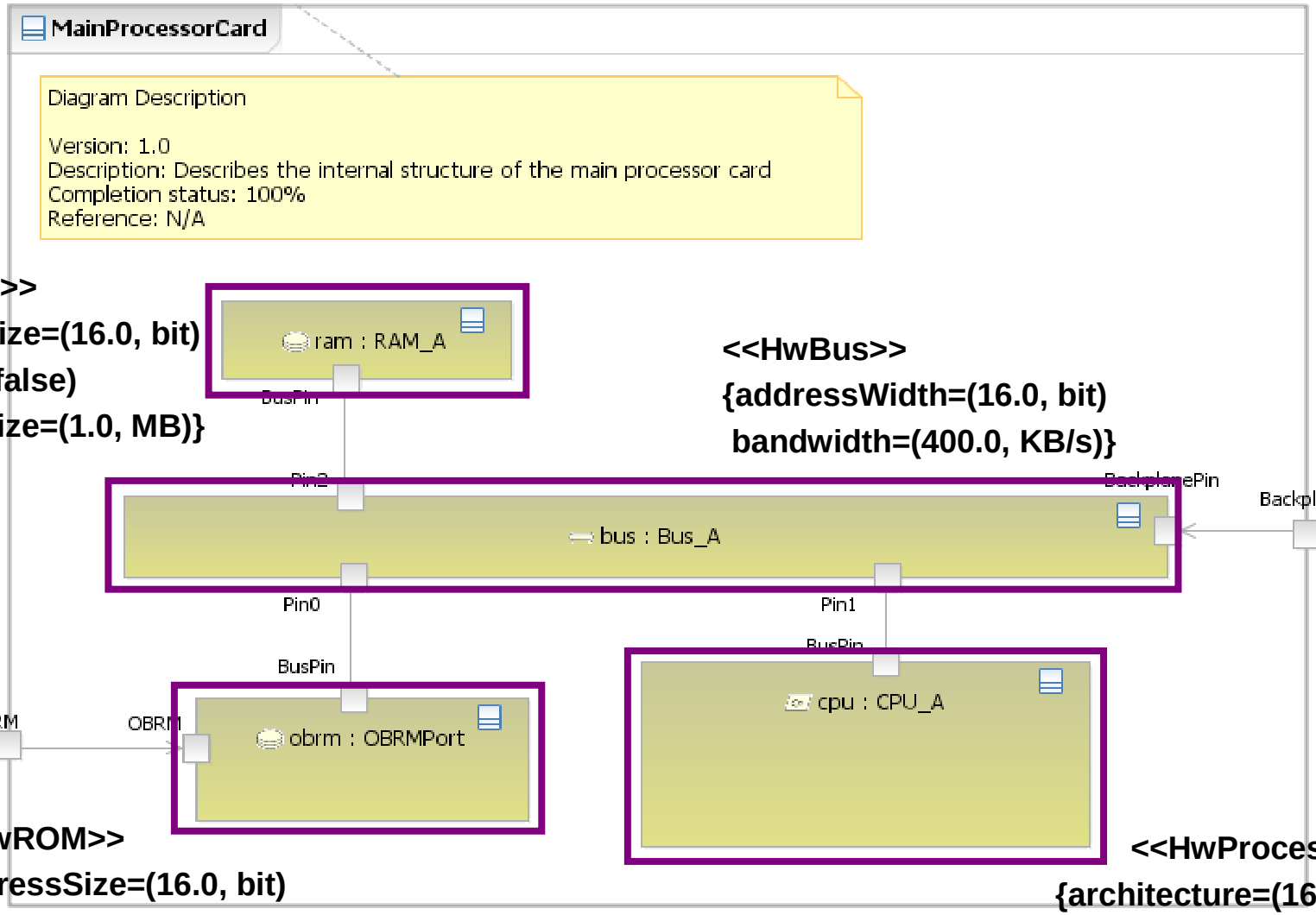


Top-level decomposition of the Flight Management and Guidance Computer

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Physical decomposition (hardware)



<<HwRAM>>
{addressSize=(16.0, bit)
isStatic=(false)
memorySize=(1.0, MB)}

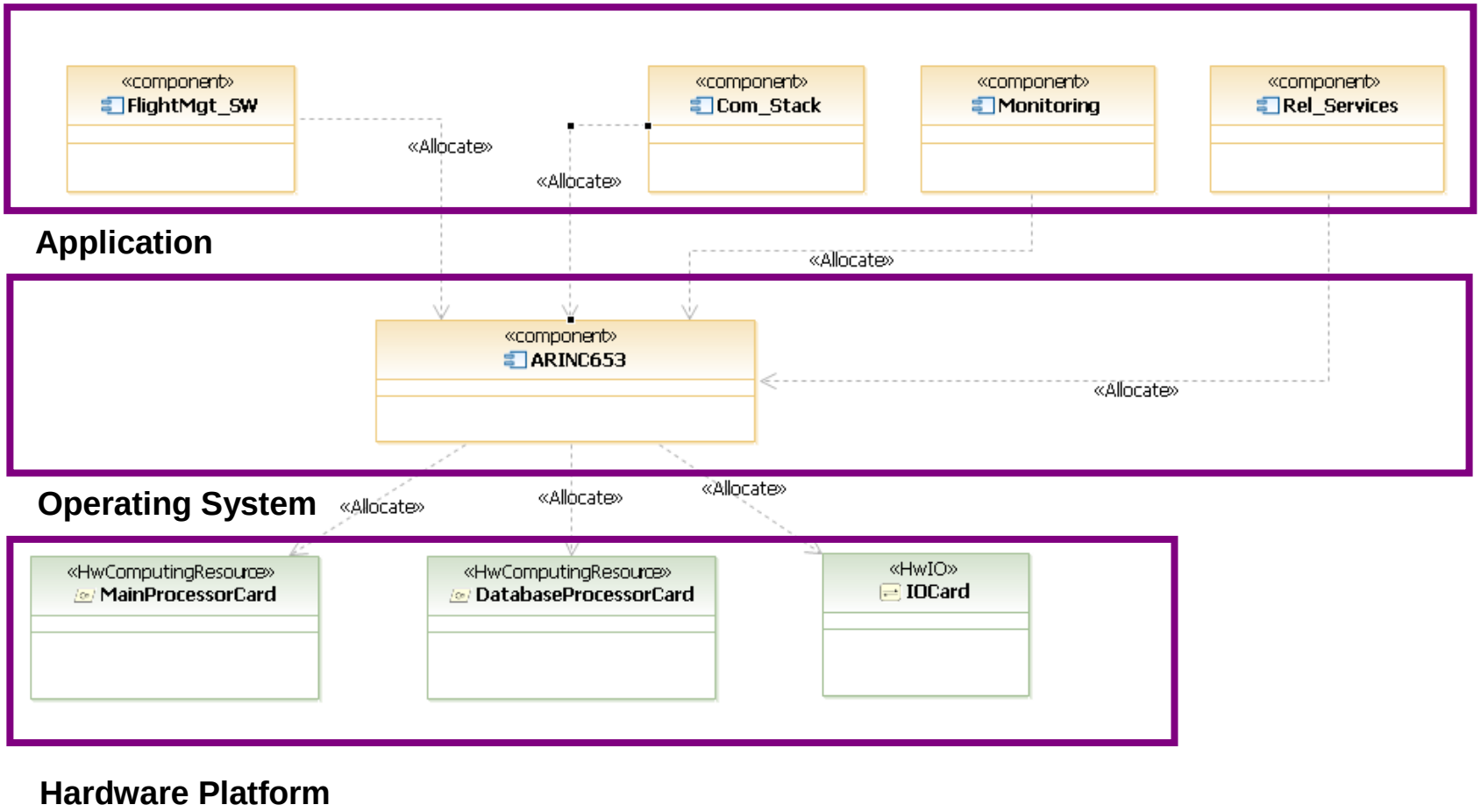
<<HwBus>>
{addressWidth=(16.0, bit)
bandwidth=(400.0, KB/s)}

<<HwROM>>
{addressSize=(16.0, bit)
memorySize=(512.0, KB)}

<<HwProcessor>>
{architecture=(16.0, bit)
frequency=(4.0, MHz)}

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet
THALES

Physical decomposition (software artifacts)



Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Experimentation highlights

System context definition

- Use case overview
- Time constraints on performance scenarios
- Types and units

System design

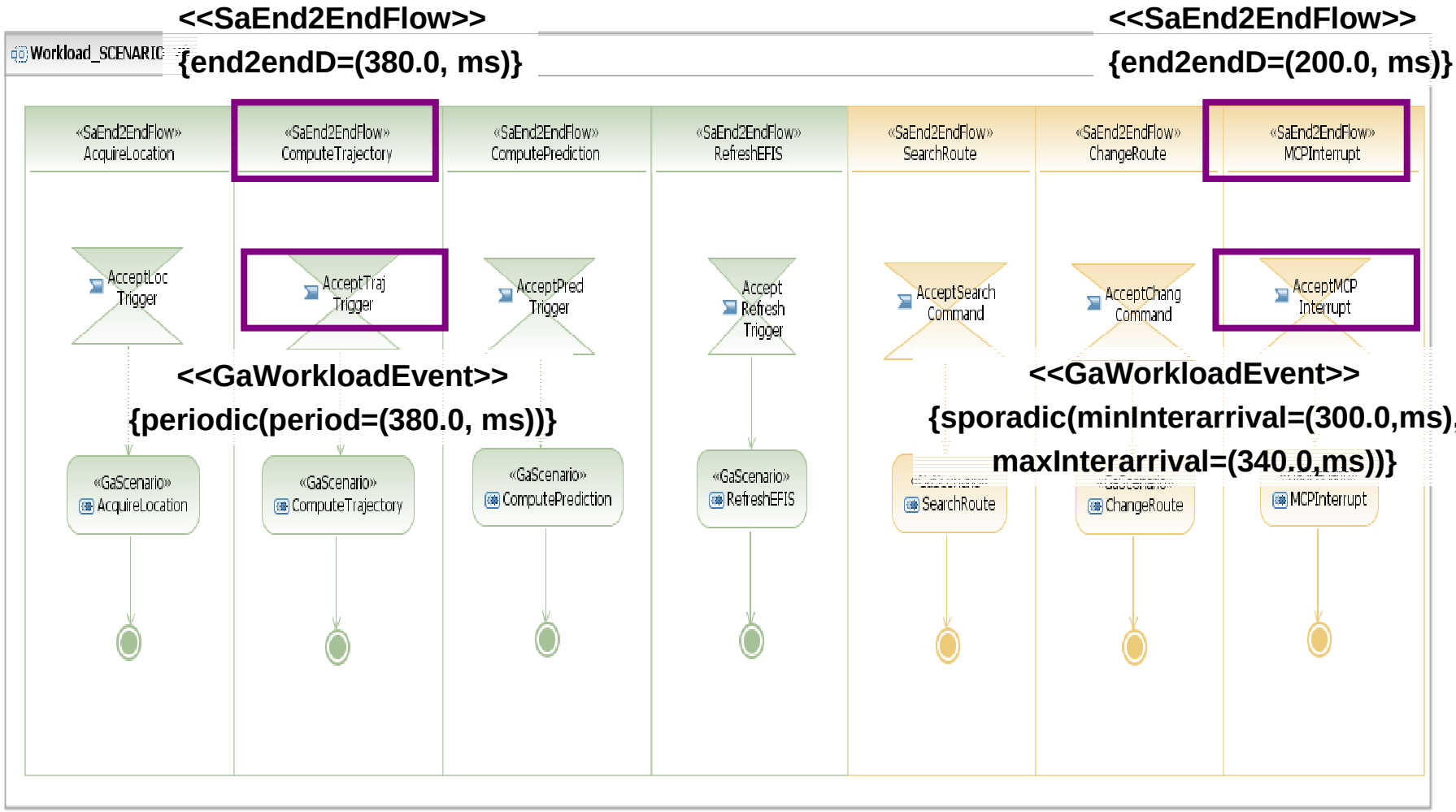
- Logical decomposition (structure, behavior)
- Physical decomposition (resources)
- Allocation

Scheduling analysis

- Building scheduling analysis model
- Validate system scheduling with RapidRMA

Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

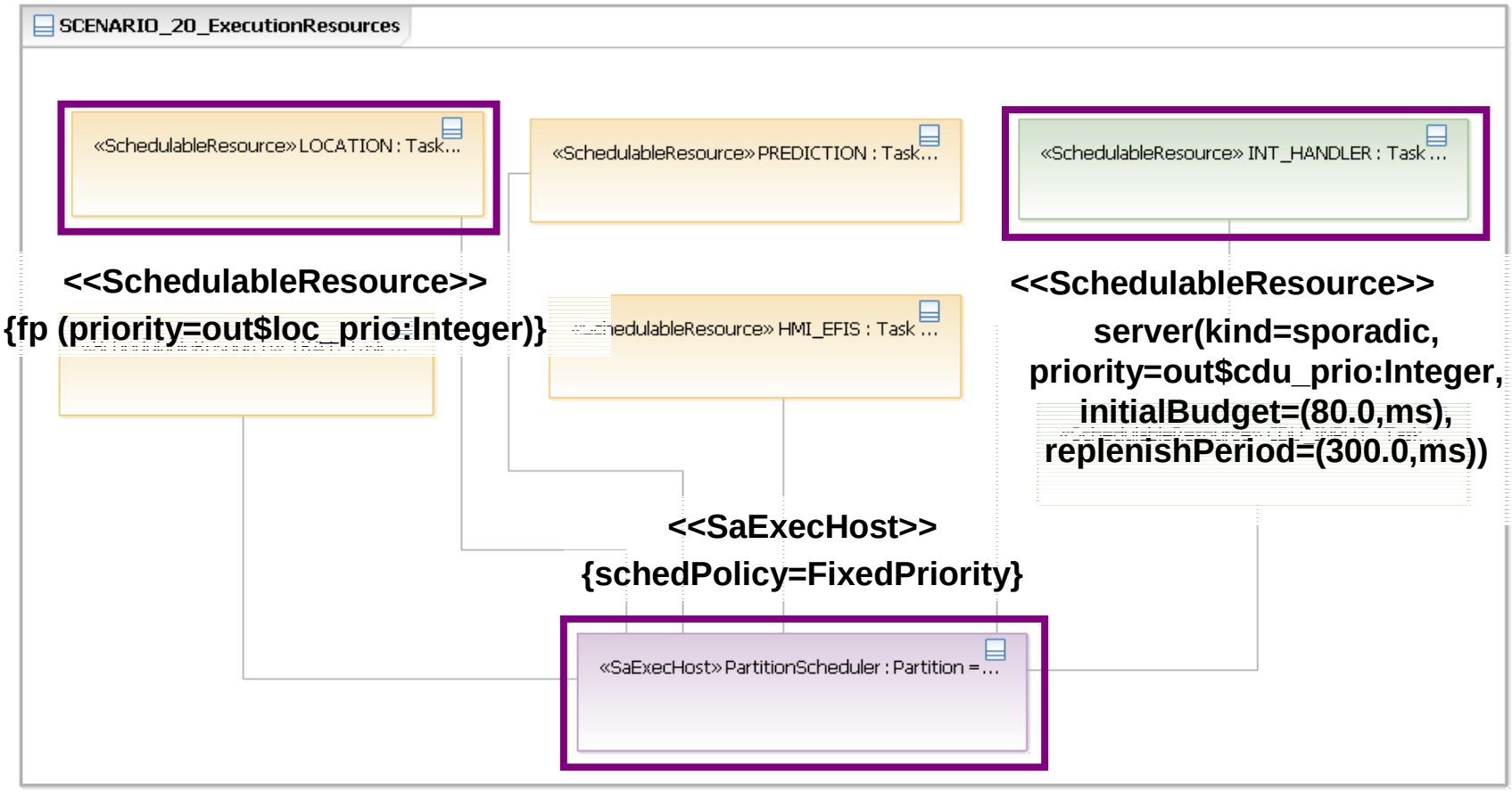
Scheduling analysis (workload)



Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

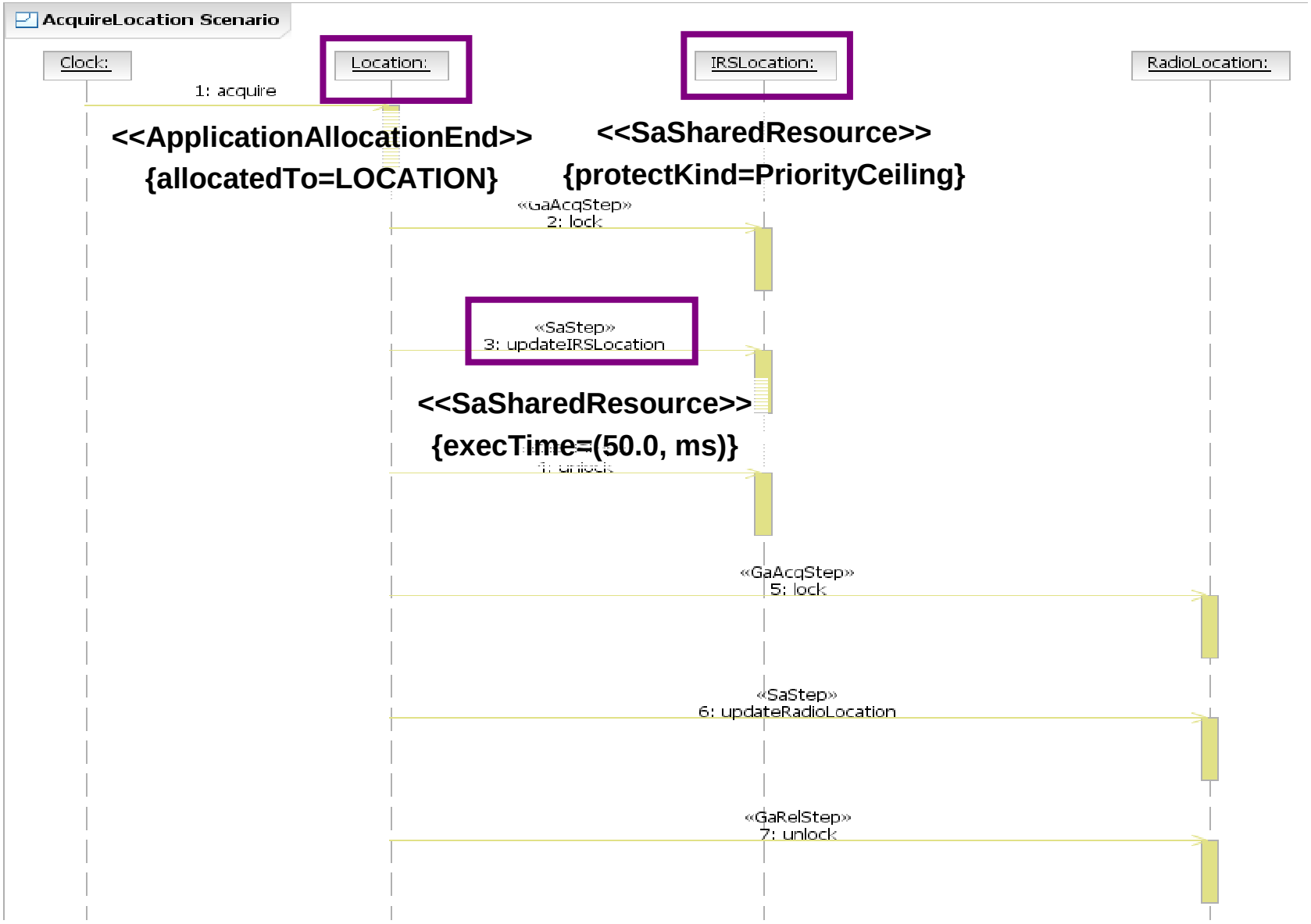
Scheduling analysis (resource)



Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

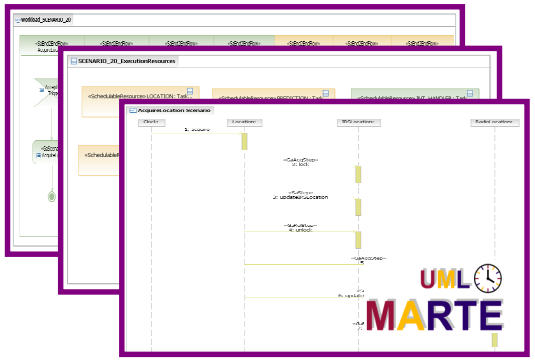
Scheduling analysis (scenario)

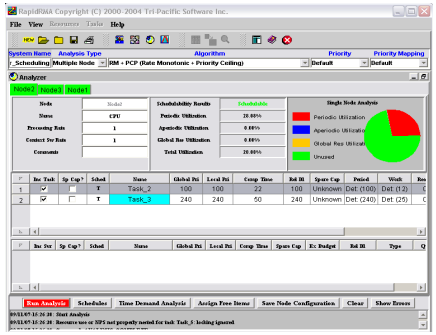


Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

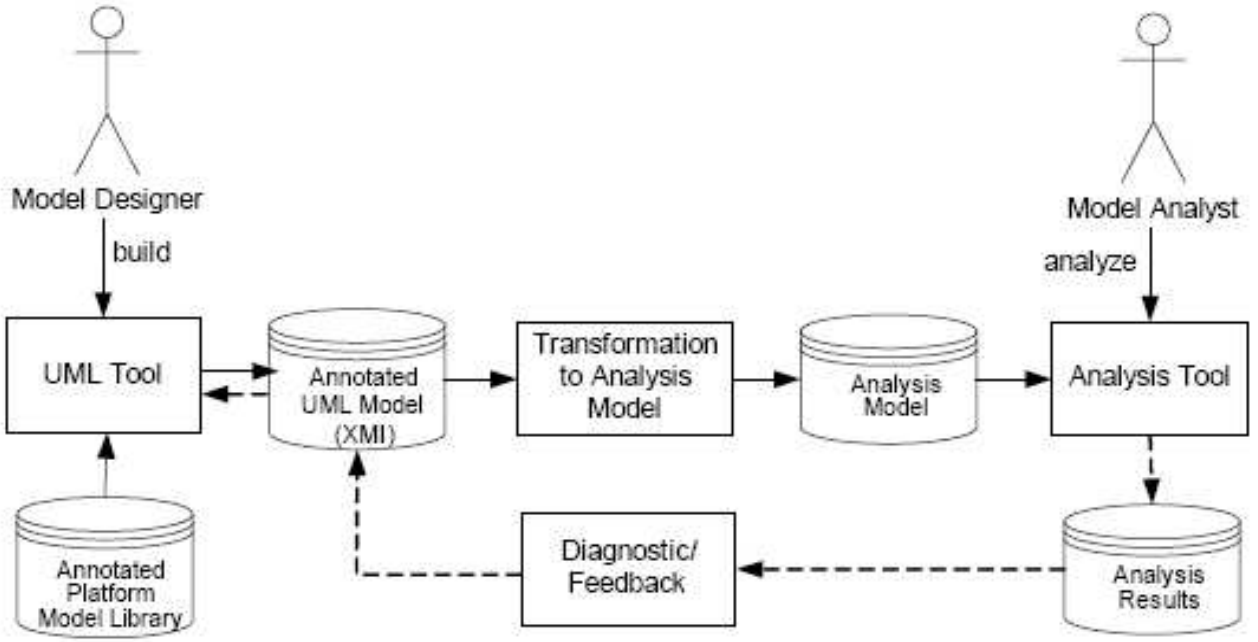
MARTE to TriPacific RapidRMA





Compute a schedule for tasks in a partition

Help dimensioning the sporadic servers



Information presented here is inspired from JAA ATPL textbooks and resources available on the Internet

THALES

Conclusion

MARTE has provided the necessary support for modeling and analyzing our system

- Modeling Time and NFPs
- SW / HW platform design
- RMA scheduling analysis

UML and MARTE are key enablers for model-based analysis and early validation

We are eager to see tool vendor implementations of MARTE