www.omgmarte.org

MARTE Tutorial

An OMG standard: UML profile to develop Real-Time and Embedded systems

Acknowledgment

- **This presentation reuses and extends material prepared by the ProMARTE partners for the OMG RTESS PTF meeting in San Diego, on March 28th 2007**

- **This tutorial has been designed in the context of CORTESS project within the CARROLL research program**
  - http://www.carroll-research.org/

- **Following persons have contributed to this tutorial**
  - CEA LIST (contact: sebastien.gerard@cea.fr)
    - Huascar Espinoza, Sébastien Gérard, Safouan Taha and Frédéric Thomas.
  - INRIA (contact: Robert.De_simone@sophia.inria.fr)
    - Charles André, Robert de Simone, Pierre Boulet
  - Thales TRT (contact: Laurent.rioux@thalesgroup.com)
    - Madeleine Faugère, Laurent Rioux, Sebastien Demathieu

How to read this tutorial

- **Within next slides, we may shown models at different levels of abstraction. We will clarify each level through following pictograms**
  - For Domain View level

  - For UML Profile View Level

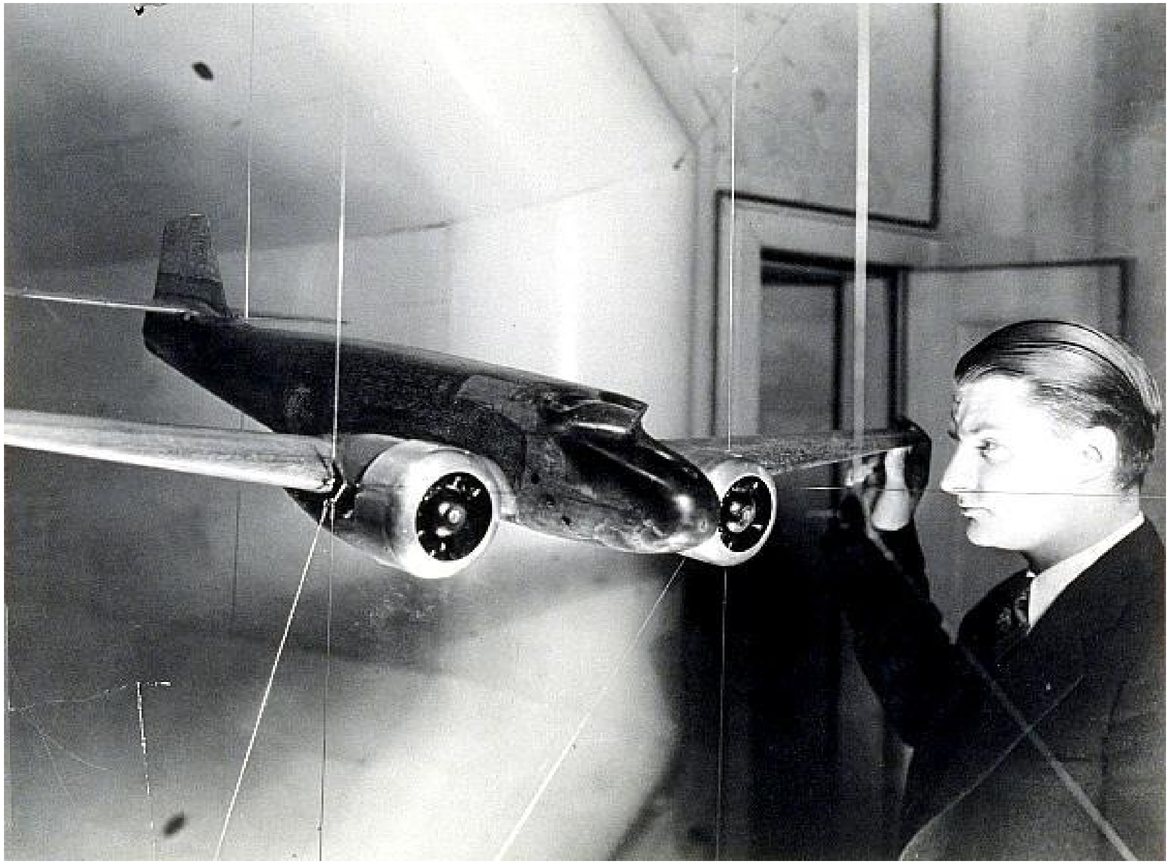
  - For User Model View Level

Agenda

- **Part 1**
  - **Introduction to MDD for RT/E systems & MARTE in a nutshell**
- **Part 2**
  - Non-functional properties modeling
  - Outline of the Value Specification Language (VSL)
- **Part 3**
  - The timing model
- **Part 4**
  - A component model for RT/E
- **Part 5**
  - Platform modeling
- **Part 6**
  - Repetitive structure modeling
- **Part 7**
  - Model-based analysis for RT/E
- **Part 8**
  - MARTE and AADL
- **Part 9**
  - Conclusions

Models in Traditional Engineering

- Probably as old as engineering



Extracted from B. Selic presentation during Summer School MDD
For DRES 2004 (Brest, September 2004)

What is a Model in MDD

Inspired from B. Selic presentation during Summer School MDD For DRES 2004 (Brest, September 2004)

- Phil Bernstein, "A Vision for Management of Complex Systems".
  A model is a complex structure that represents a design artifact such as a relational schema, an interface definition (API), an XML schema, a semantic network, a UML model or a hypermedia document.
- OMG, "UML Superstructure".
  A model captures a view of a physical system. It is an abstraction of the physical system, with a certain purpose. This purpose determines what is included in the model and what is relevant. Thus the model completely describes those aspects of the physical system that are relevant to the purpose of the model, at the appropriate level of detail.
- OMG, "MDA Guide".
  A formal specification of the function, structure and/or behavior of an application or system.
- Steve Mellor, et al., "UML Distilled"
  A model is a simplification of something so we can view, manipulate, and reason about it, and so help us understand the complexity inherent in the subject under study.
- Anneke Kleppe, et. al. "MDA Explained"
  A model is a description of (part of) a system written in a well-defined language. A well-defined language is a language with well-defined form (syntax), and meaning (semantics), which is suitable for automated interpretation by a computer.
- Chris Raistrick et al., "Model Driven Architecture with Executable UML"
  A formal representation of the function, behavior, and structure of the system we are considering, expressed in an unambiguous language.
- J. Bézivin & O. Gerbé, "Towards a Precise Definition of the OMG/MDA Framework"
  A simplification of a system built with an intended goal in mind; The model should be able to answer questions in place of the actual system.

One definition

- A reduced/abstract representation of some system that highlights the properties of interest from a given point of view.
- The point of view defines concern and scope of the model.

The Model

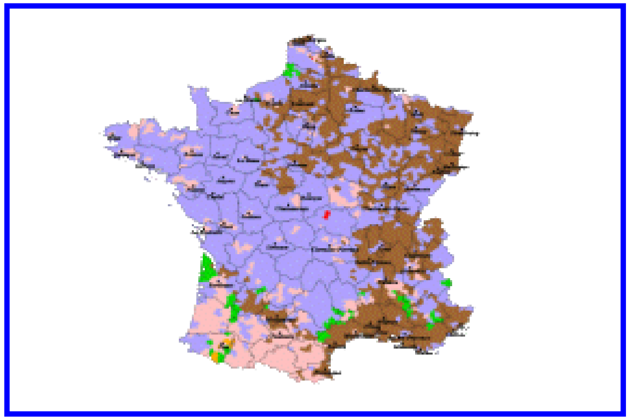
- Map is based on a legend (explicit or implicit)
  - Here the map of bicycle roads of Seattle
- As a map, the legend is defined in a graphical language, it means also the legend is declared with a similar formalism.
- If the Map is a Model, the legend is the meta-model defining the subset of graphical language used to build the model
- The Legend is necessary to interpret the map.
- If the legend is not shown, this mean we refer to a standard legend and implicit.

The Legend

From J. Bezivin / INRIA

A Model without its meta-model has no meaning

Candidates at the Presidential election In France in 2002

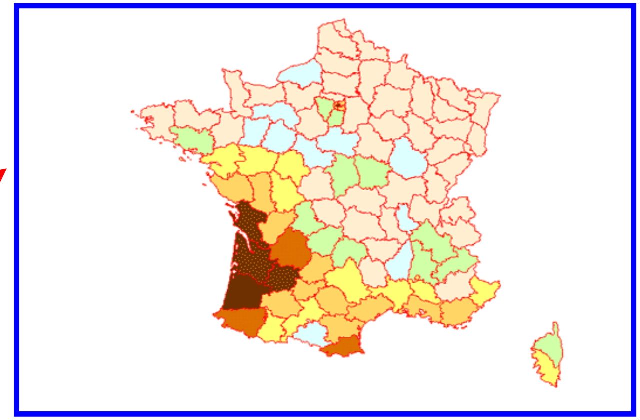


- Principales villes françaises
- Limites départementales

Candidat arrivé en tête au premier tour

- Chirac
- Le Pen
- Jospin
- Bayrou
- Chevènement
- Saint-Josse
- Hue
- Mégret

Percentage Of town infested of termites

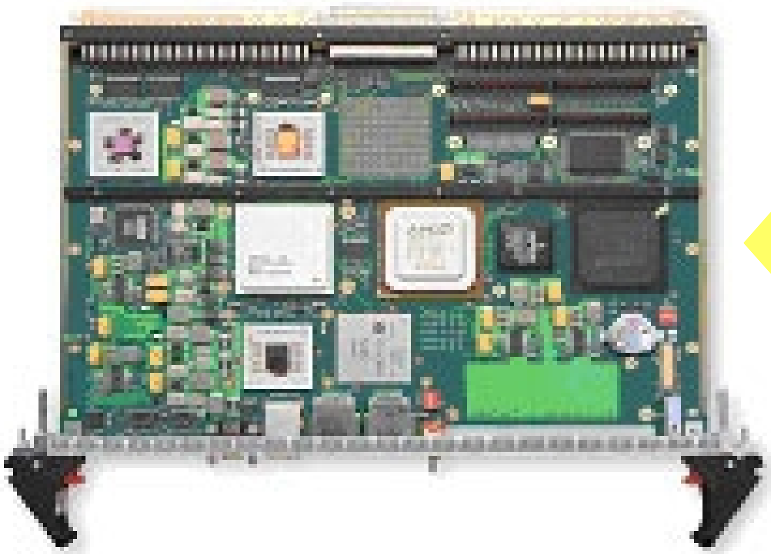


- de 75 à 100
- de 50 à 75 %
- de 25 à 50 %
- de 10 à 25 %

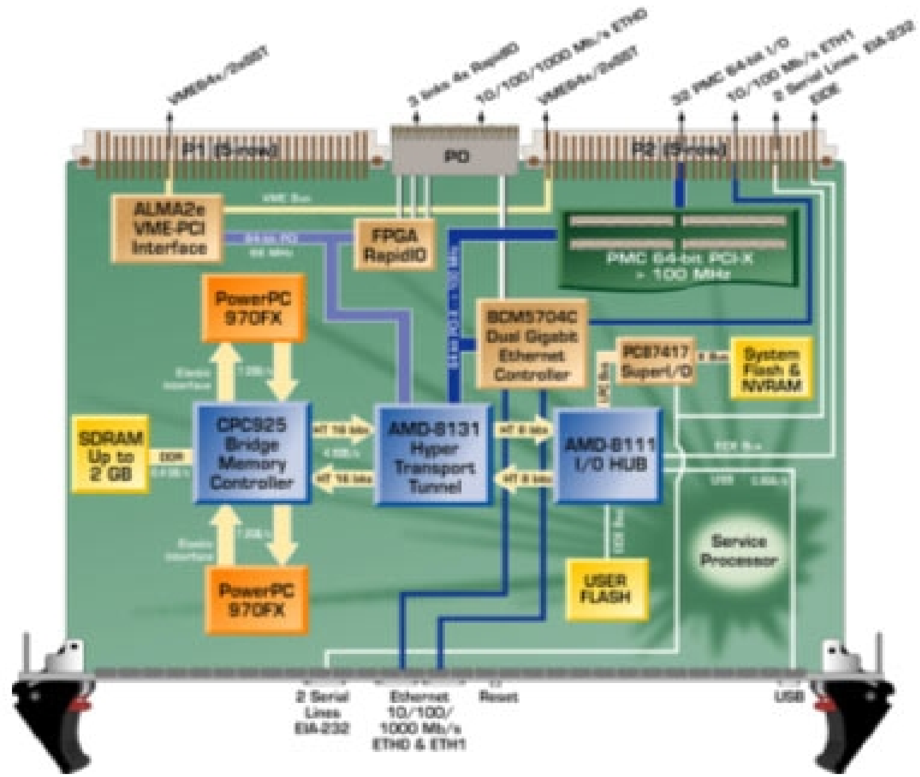
From J. Bezivin / INRIA

UML MARTE www.omgmarte.org

The Model help to understand the system



System to model



Functional model

- For Functional viewpoint and its design

Why Model Driven Engineering is Needed?

- **To deal with complexity of systems development**
  - Abstract a problem to focus on some particular points of interest
    ➔ improve understandability of a problem
  - Possible set of nearly independent views of a model
    - Separation of concerns (e.g. “Aspect Oriented Modeling”)
  - Iterative modeling may be expressed at different level of abstraction

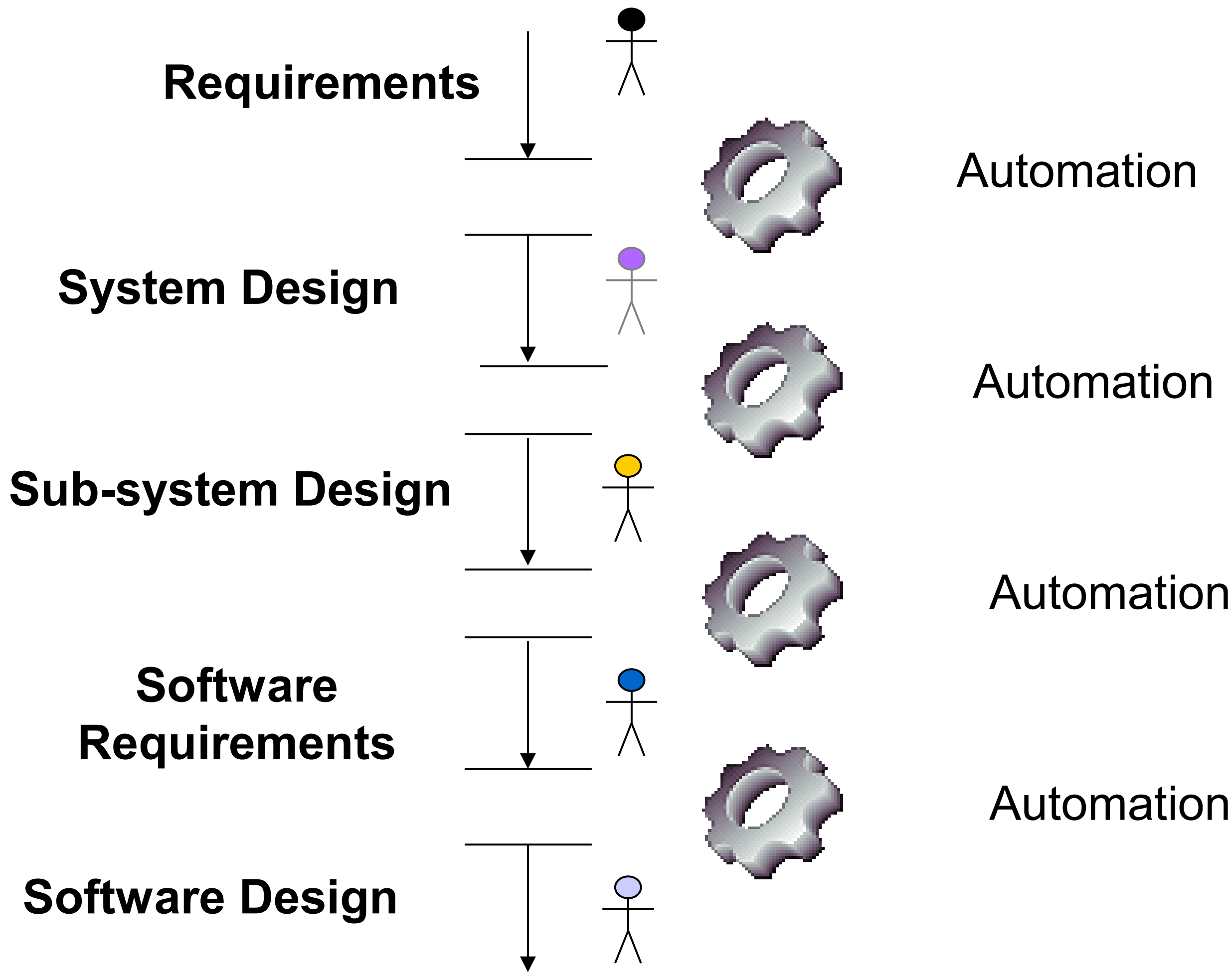
- **To minimize development risks**
  - Through analysis and experimentation performed early in the design cycle
  - Enable to investigate and compare alternative solutions

- **To improve communication ...**
- **... to foster information sharing and reuse!**
  - A good model is better than a long speech !

- **To reduce development flaws**
  - Automatic model transformation is less error-prone than building a specific compiler

Extracted from S.Gerard (ECRTS07)

Why: Provide Continuum in development process



Characteristics of Useful Models

- **Abstract**
  - Emphasize important aspects while removing irrelevant ones
- **Understandable**
  - Expressed in a form that is readily understood by observers
- **Accurate**
  - Faithfully represents the modeled system
- **Predictive**
  - Can be used to answer questions about the modeled system
- **Inexpensive**
  - Much cheaper to construct and study than the actual system

To be useful, engineering models must satisfy <u>all</u> of these characteristics!

(Extracted from B. Selic presentation during Summer School MDD For DRES 2004 (Brest, September 2004)

A Bit of Modern Software…

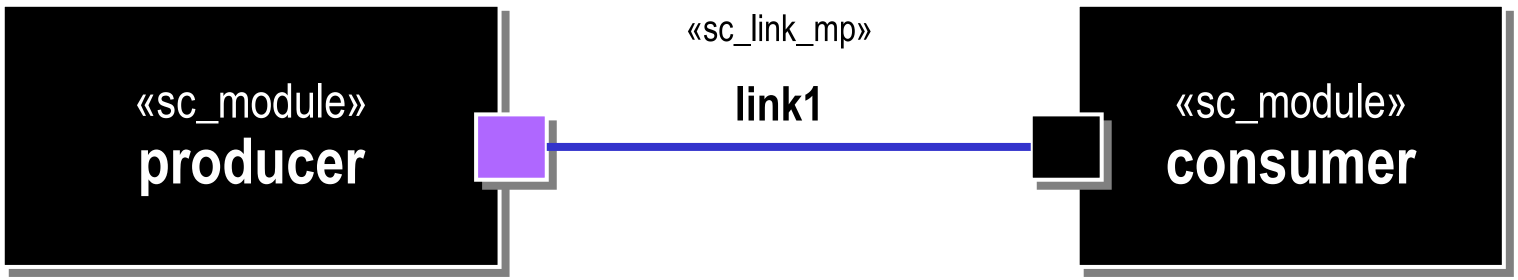
```
SC_MODULE(producer) {
  sc_outmaster<int> out1;
  sc_in<bool> start; // kick-start
  void generate_data () {
    for(int i =0; i <10; i++) {
      out1 =i ; //to invoke slave;
    }
  }
  SC_CTOR(producer) {
    SC_METHOD(generate_data);
    sensitive << start;
  }
};
SC_MODULE(top) { // container
  producer *A1;
  consumer *B1;
  sc_link_mp<int> link1;
  SC_CTOR(top) {
    A1 = new producer("A1");
    A1.out1(link1);
    B1 = new consumer("B1");
    B1.in1(link1);
  }
};
```

```
SC_MODULE(consumer) {
  sc_inslave<int> in1;
  int sum; // state variable
  void accumulate (){
    sum += in1;
    cout << "Sum = " << sum << endl;
  }
  SC_CTOR(consumer) {
    SC_SLAVE(accumulate, in1);
    sum = 0; // initialize
  }
};
```

Can you spot the architecture?

(Extracted from B. Selic presentation during Summer School MDD For DRES 2004 (Brest, September 2004)

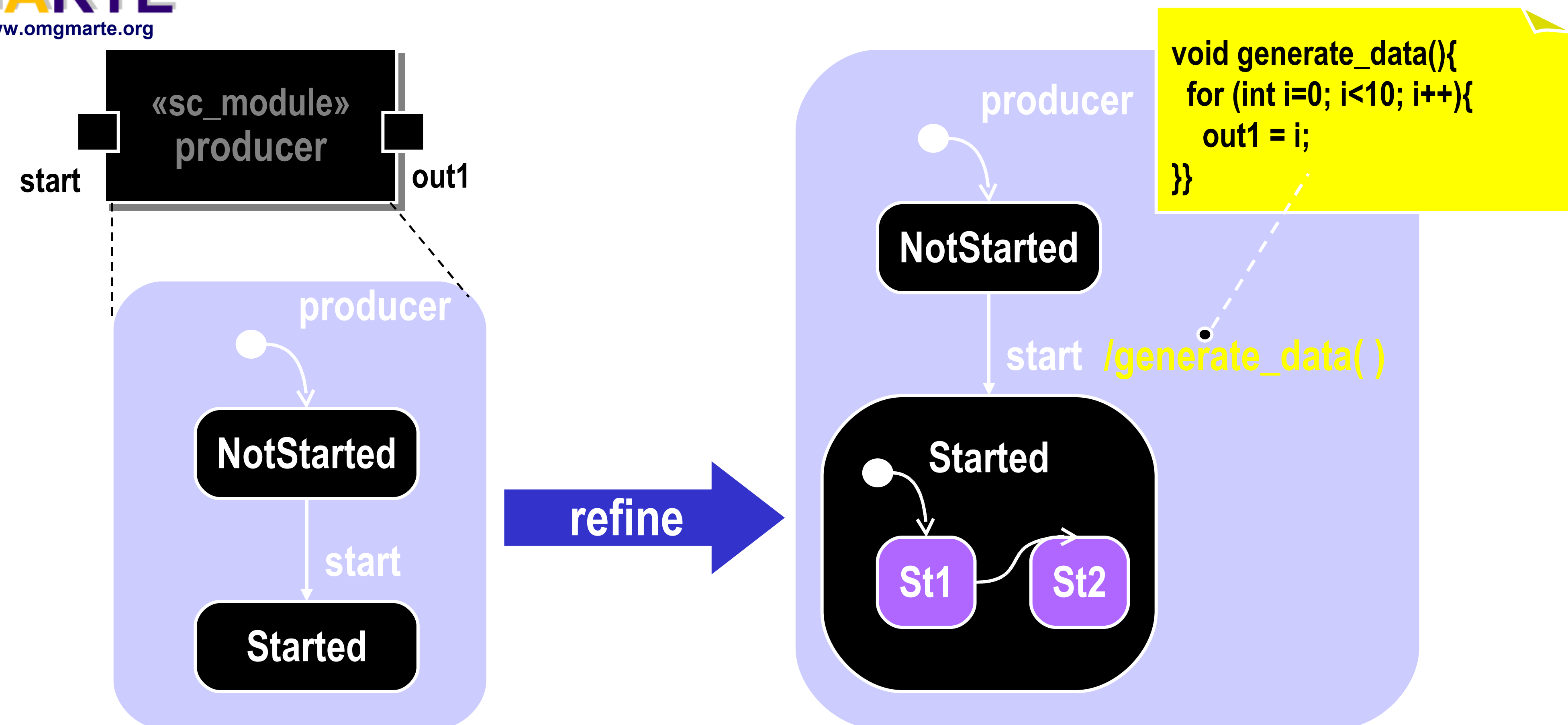
...and its UML Model



Can you spot the architecture?

(Extracted from B. Selic presentation during Summer School MDD For DRES 2004 (Brest, September 2004))

Model Evolution: Refinement

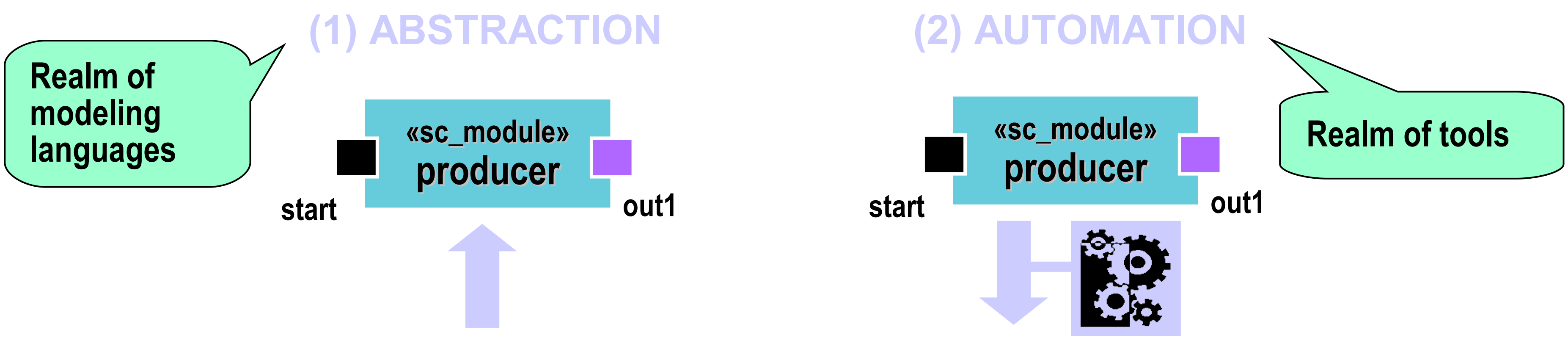


- Models can be refined continuously until the specification is complete

(Extracted from B. Selic presentation during Summer School MDD For DRES 2004 (Brest, September 2004)

Model-Driven Style of Development (MDD)

- An approach to develop systems and softwares in which the focus and primary artifacts of development are models (as opposed to programs)
- Based on two time-proven methods




```
SC_MODULE(producer){
  sc_inslave<int> in1;
  int sum; //
  void accumulate () {
    sum += in1;
    cout <<   Sum =
               << sum <<
endl;}
```

```
SC_MODULE(producer) {
  sc_inslave<int> in1;
  int sum; //
  void accumulate (){
    sum += in1;
    cout <<   Sum =
            << sum <<
endl;}
```

(Extracted from B. Selic presentation during Summer School MDD For DRES 2004 (Brest, September 2004)

Profiling UML for a Domain

- **Advantages of UML Profiles**
  - Reuse of language infrastructure (tools, specifications)
  - Require less language design skills
  - Allow for new (graphical) notation of extended stereotypes
  - A profile can define model viewpoints
    - E.g., UML activity diagram extended to specify multitask behavior

- **Disadvantage**
  - Constrained by UML metamodel

Extracted from S.Gerard (ECRTS07)

UML2 Extension Mechanisms

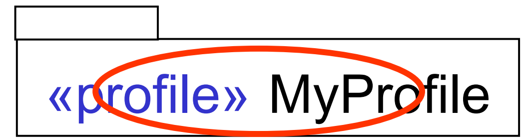
- ## Profiles
  - Define limited extensions to a reference metamodel with the purpose of adapting the metamodel to a specific platform or domain.
  - Consists of stereotypes that extend the metamodel classes (metaclasses).

- ## Stereotypes
  - Define how a specific metaclass may be extended
  - Provide additional semantics information, but only for:
    - Semantics restriction or clarification of existing concept
    - New features (but compatible with exiting one!)
  - Ensure introduction of domain specific terminology
    - E.g., EAST-ADL2, a UML profile for automotive ECUs (http://www.atesst.org)
    - May define specific notation
      - **E.g., new icons or shapes**
  - May have values that are usually referred to as tagged values

Extracted from S.Gerard
(ECRTS07)

Profile Notation

- **Profile is a stereotyped package**



- **Applying a profile**
  - All extensions are then available for modeling

  - If multiple profiles are applied:
    - Referenced MMs have to be identical…
      … and the model has also to refer the same MM.
    - Their constraint sets do not have to conflict
    - In case of naming conflict, use namespace notation
      - **<ProfileName>::<StereotypeName>**
      - **e.g. «MyProfile1::name» & «MyProfile2::name»**

Extracted from S.Gerard (ECRTS07)

The Profile Concept (cont.)

- **A profile package may import external packages**
  - "Normal" packages
    - e.g. external pkgs defining specific types for a profile

  - "Profile" packages

  - All imported elements may be used in pkgs applying the profile

Extracted from S.Gerard (ECRTS07)

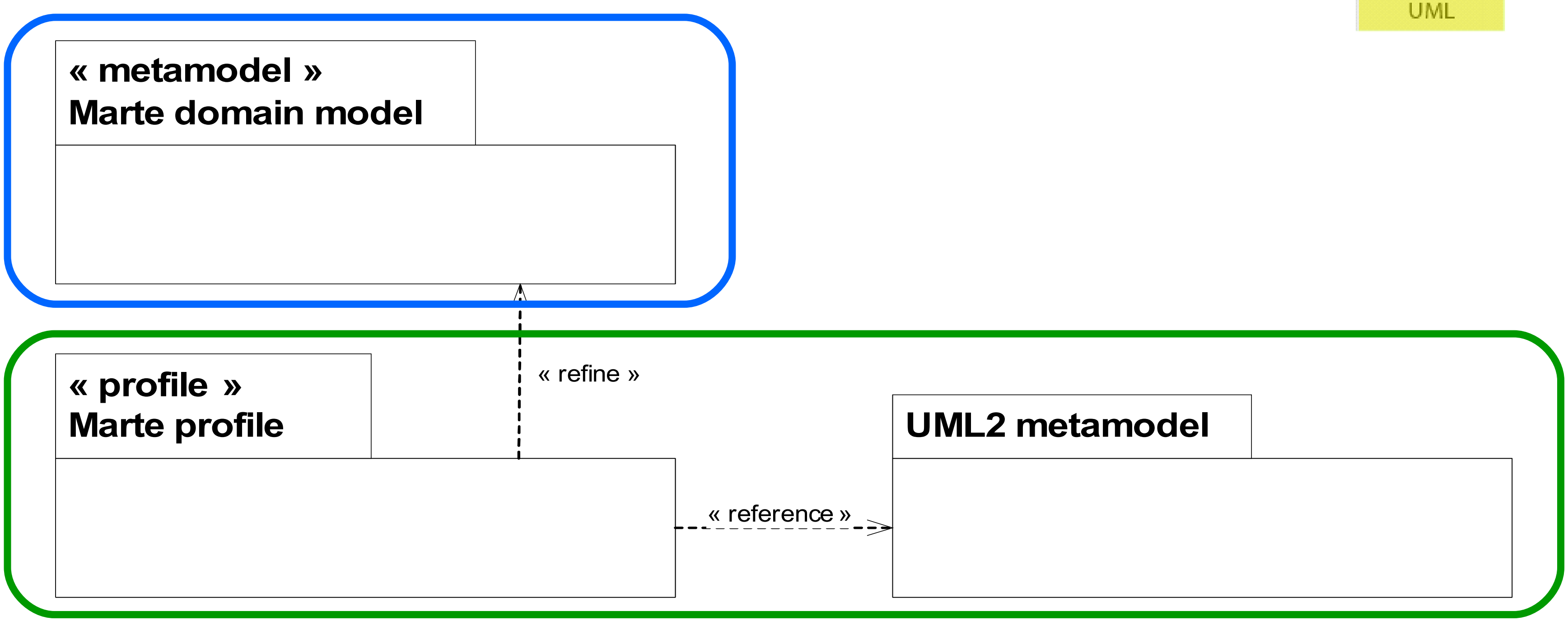
Design Pattern Adopted for the MARTE Profile

- **Stage 1 → Description of MARTE** domain models (Domain View)
  - Purpose: Formal description of the concepts required for MARTE
  - Techniques: Meta-modeling

Domain

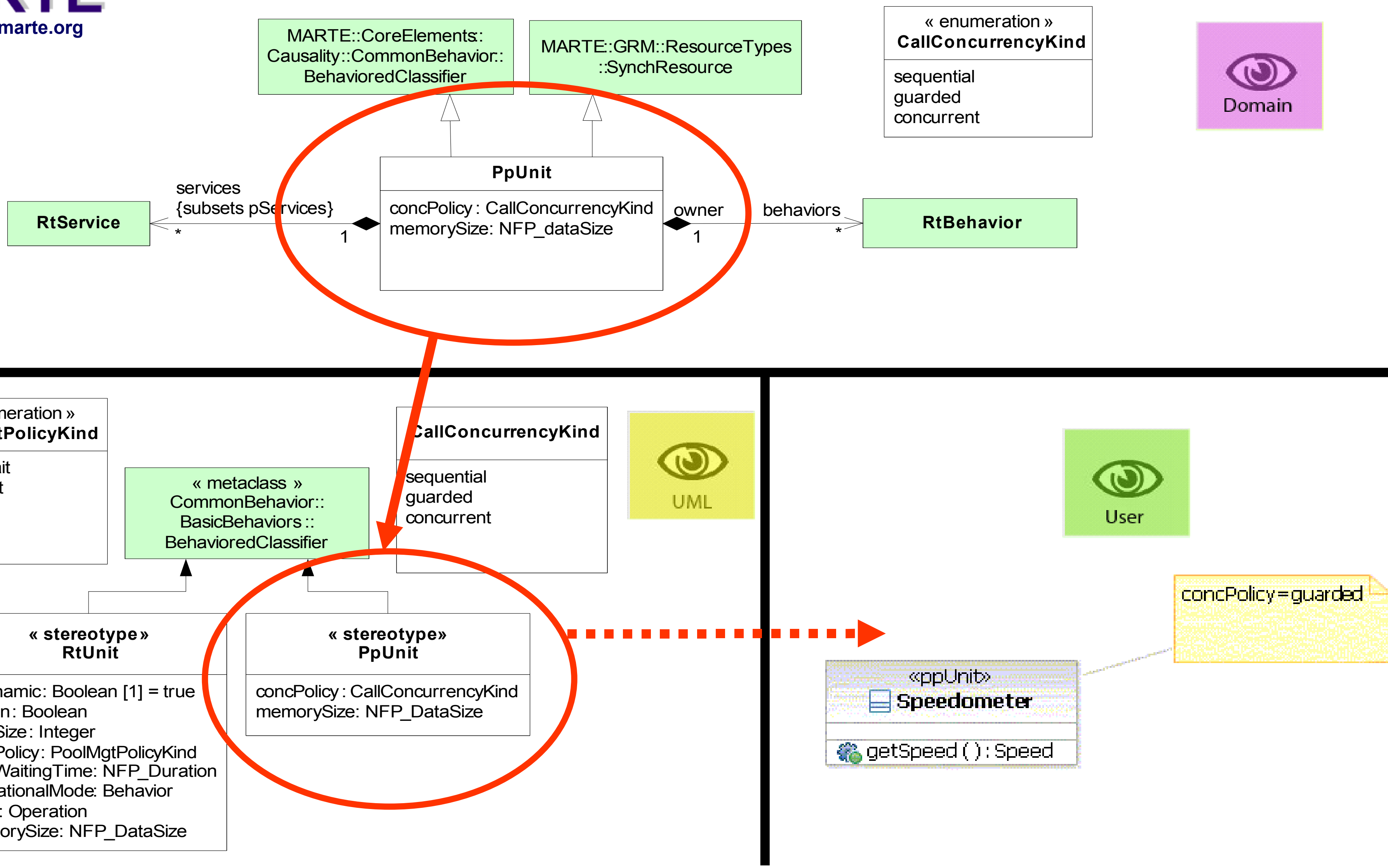
- **Stage 2 → Mapping of MARTE domain models towards UML2:** (UML Representation)
  - Purpose: MARTE domain models design as a UML2 extensions
  - Techniques: UML2 profile

UML



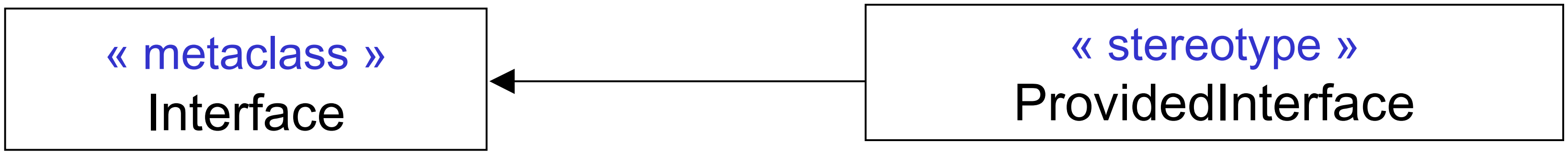
Extracted from S.Gerard (ECRTS07)

Example: Domain model → Profile → Usage



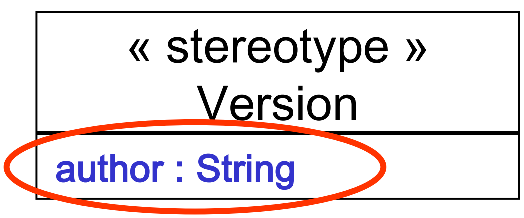
Notation for Stereotype Definition (Uml Representation)

- **Stereotype definition**



- **Required stereotype**
  - Extended meta-class may only be instantiated under its stereotyped form

- **Stereotype properties**



Extracted from S.Gerard (ECRTS07)

Notation for Stereotype Usage (user model-level)

- **Applying a stereotype**

« providedInterface »
MyInterface

User

- **Applying several stereotypes**

« providedInterface, version »
MyInterface

or

« providedInterface», «version »
MyInterface

- **Specifying values of a stereotype**

«version»
MyInterface

author = “myname”

User

  - Use name of stereotypes when possible confusion

«version, status»
MyClass

«version»
author = “myname”
«status»
value = tested

User

Extracted from S.Gerard (ECRTS07)

UML Profiles for RTES

Extracted from S.Gérard (ECRTS07)

SPT
(2005-adopted)

Tool Support

Time Analysis Support

- **SPT was the first OMG's UML profile for Real-Time Systems:**
  - Support for **S**chedulability Analysis with RMA-type techniques
  - Support for **P**erformance Analysis with Queuing Theory and Petri Nets
  - A rich model for "metric" **T**ime and Time Mechanisms
- **Several improvements were required:**
  - Modeling HW and SW platforms, Logical Time, MoCCs, CBSE…
  - Alignment to UML2, QoS&FT, MDA,…
  - SPT constructs were considered too abstract and hard to apply
  - …

Hence, a Request For Proposal for a new profile was issued.

The ProMARTE Team

- **Industrials**
  - Alcatel*
  - Lockheed Martin*
  - Thales*
  - France-Telecom
- **Tool vendors**
  - ARTISAN Software Tools*
  - International Business Machines*
  - Mentor Graphics Corporation*
  - Softeam*
  - Telelogic AB (I-Logix*)
  - Tri-Pacific Software
  - France Telecom
  - No Magic
  - Mathworks
- **Academics**
  - Carleton University
  - Commissariat à l'Energie Atomique
  - ESEO
  - ENSIETA
  - INRIA
  - INSA from Lyon
  - Software Engineering Institute (Carnegie Mellon University)
  - Universidad de Cantabria

Public website:

www.omgmarte.org

* Submitter to OMG UML Profile for MARTE RFP

Relationships with other OMG Standards

Relationships with generic OMG standards

- Profile the UML2 superstructure meta-model
- Replace UML Profile for SPT (Scheduling, Performance and Time)
- Use OCL2 (Object Constraints Language)

Relationships with RT&E specific OMG standards

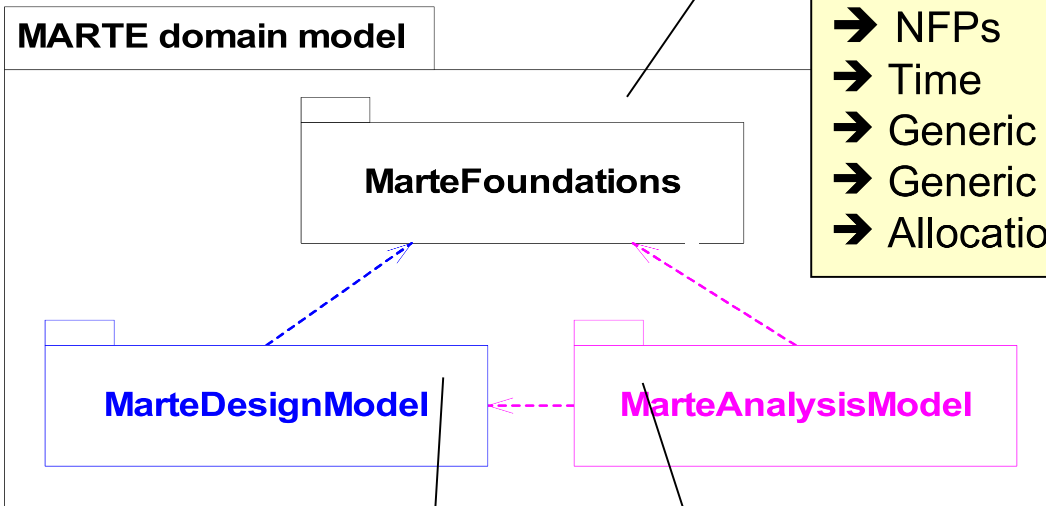
- Existing standards
  - The UML profile for Modeling QoS and FT Characteristics and Mechanisms
    - **Addressed through MARTE NFP package (in a way detailed in the NFP presentation)**
  - The UML profile for SoC (System On Chip)
    - **More specific than MARTE purpose**
  - The Real-Time CORBA profile
    - **Real-Time CORBA based architecture can be annotated for analysis with Marte**
  - The UML profile for Systems Engineering (SysML)
    - **Specialization of SysML allocation concepts and reuse of flow-related concepts**
    - **Ongoing discussion to include VSL in next SysML version**
    - **Overlap of team members**

MARTE Overview

UML MARTE
www.omgmarte.org

Foundations for RT/E systems modeling and analysis:
- ➔ CoreElements
- ➔ NFPs
- ➔ Time
- ➔ Generic resource modeling
- ➔ Generic component modeling
- ➔ Allocation

Domain



Specialization of MARTE foundations for modeling purpose (specification, design, …):
- ➔ RTE model of computation and communication
- ➔ Software resource modeling
- ➔ Hardware resource modeling

Specialization of foundations for annotating model for analysis purpose:
- ➔ Generic quantitative analysis
- ➔ Schedulability analysis
- ➔ Performance analysis

Extracted from S.Gerard (ECRTS07)