



Towards a traceability model in a MARTE-based methodology for real-time embedded systems

Hung LE DANG, [Hubert DUBOIS](#) and Sébastien GERARD

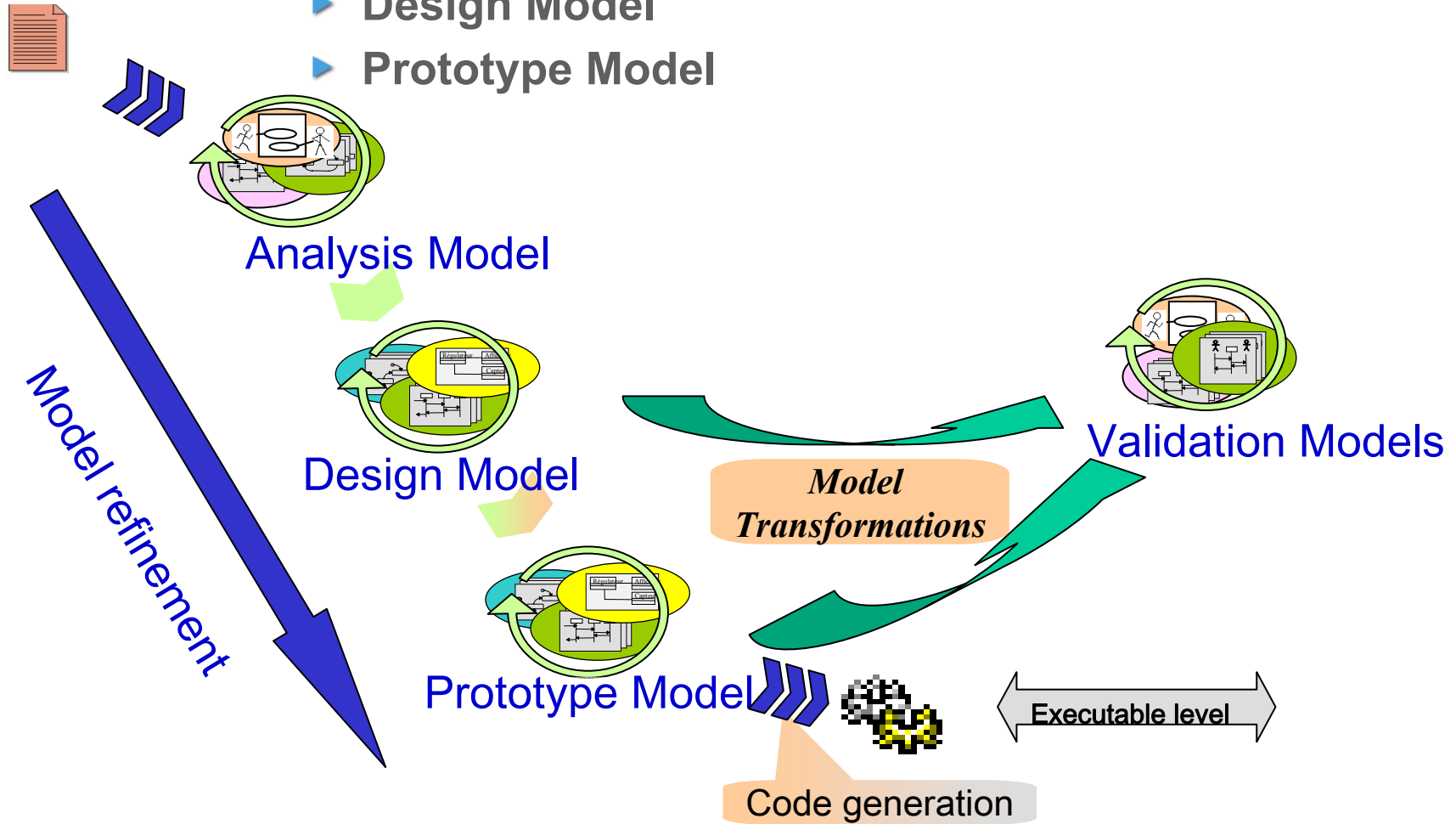
CEA LIST, 91191 Gif sur Yvette, France

e-mail: {Hung.Le-Dang, Hubert.Dubois, Sebastien.Gerard}@cea.fr)



Requirements

- From requirements to the solution
 - ▶ Analysis Model
 - ▶ Design Model
 - ▶ Prototype Model



Model transformation & traceability

➤ Real-time specificities

- Appear earlier in the requirements
- Have to be integrated in the models
- Have to be validated and verified

✗ Lacks in model-based approaches

- ✗ Several model transformations and high complexity for coconsidering traceability of model elements & properties
- ✗ Traceability for:
 - ✗ requirements,
 - ✗ model elements & properties,
 - ✗ transformation towards specific tools (for V&V for instance) or code & feedbacks

⇒ The purpose = an integrated approach for:

- Supporting requirements in the UML models
- Supporting traceability during model transformation/refinement
- Supporting V&V of model transformations



Plan



- Introduction
- Traceability management in Accord|UML
- Requirement definition and traceability
- Traceability in the models with MARTE
- Example
- Traceability in the models for verification support
- Methodological aspects
- Conclusion & perspectives



Traceability management in Accord_{UML}

- **From requirements to models**
 - ▶ A clear association between requirements and models
 - ▶ To guaranty the validation of the requirements
 - ▶ To help in the verification of the requirements
 - ▶ To help in certification processes
- **From Analysis/Design Models to Validation Models**
 - ▶ To clearly associate model elements in the different modeling steps
 - ▶ To help in model feedbacks when analyzing the V&V results
 - ▶ Integration in model transformation
- **From transformations to verification process**
 - ▶ Connexion to V&V techniques
 - ▶ Verification of the transformation [→ LeDang-Dubois-Gérard, iFM'09]

• SysML: a profile for requirement modeling

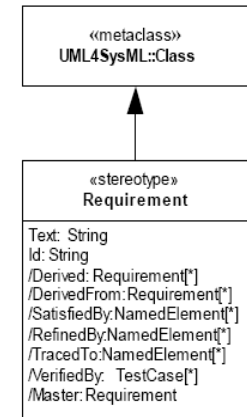
- ▶ requirements as first class concepts in UML
- ▶ Traceability concerns:
 - ▶ Requirements ↔ Requirements (e.g. *Derive*)
 - ▶ Requirements ↔ Model Elements (e.g. *Satisfy*, *Verify*)

→ Not sufficient for:

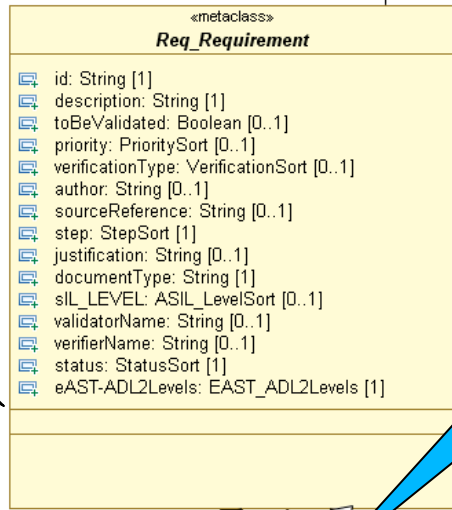
- Requirements traceability in an heterogeneous context (AUTOSAR, Simulink, etc...)
- Requirements validation & verification informations (specific tools, tests, etc...)
- Requirements management (precise identification, classification, reports, etc...)
- Requirement diversity (not only textual requirements)

⇒ We adapt this standard to our needs: a Requirement meta-model

- ⇒ Consideration of heterogeneous formalisms (UML & others)
- ⇒ Requirement definition
- ⇒ Requirement traceability

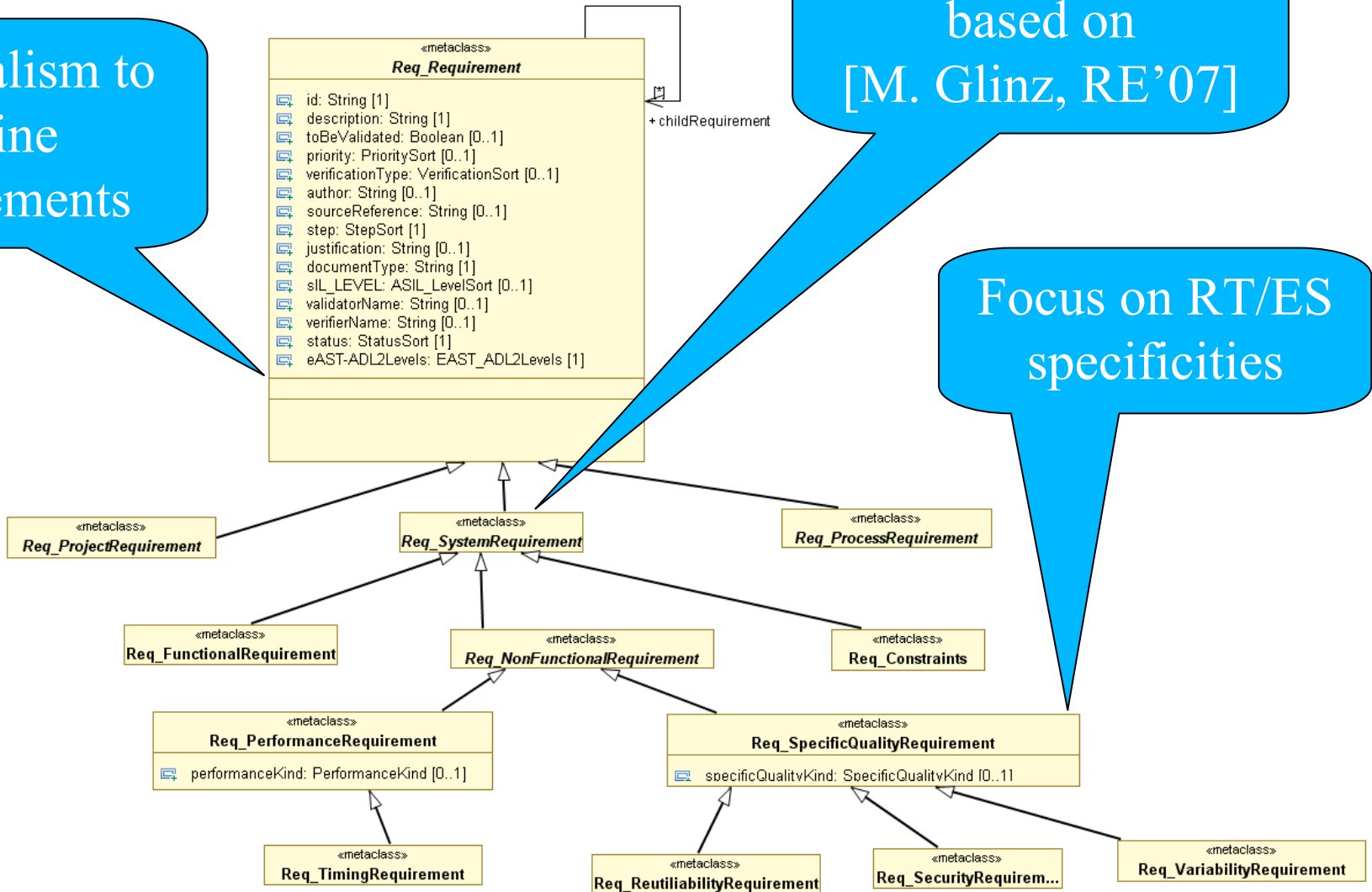


A formalism to define requirements



A classification based on [M. Glinz, RE'07]

Focus on RT/ES specificities



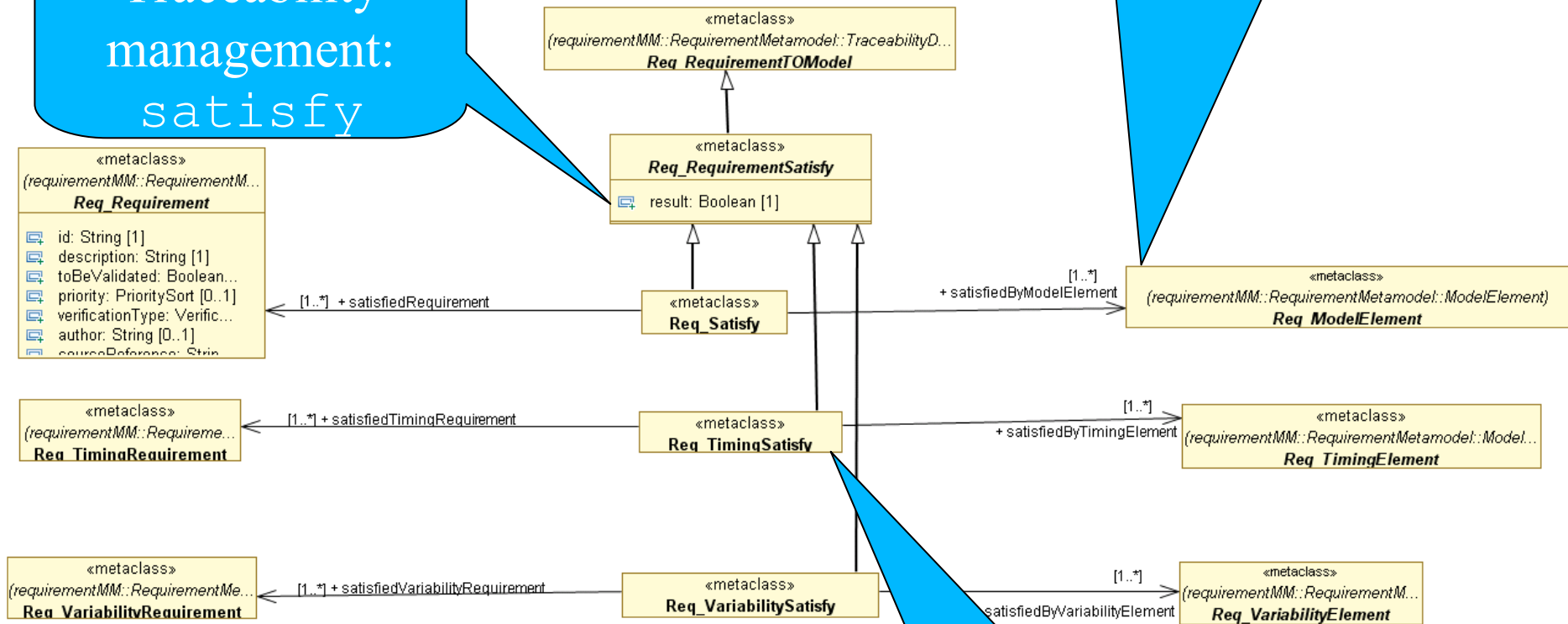
Requirement Traceability – III / IV



list

Traceability management: satisfy

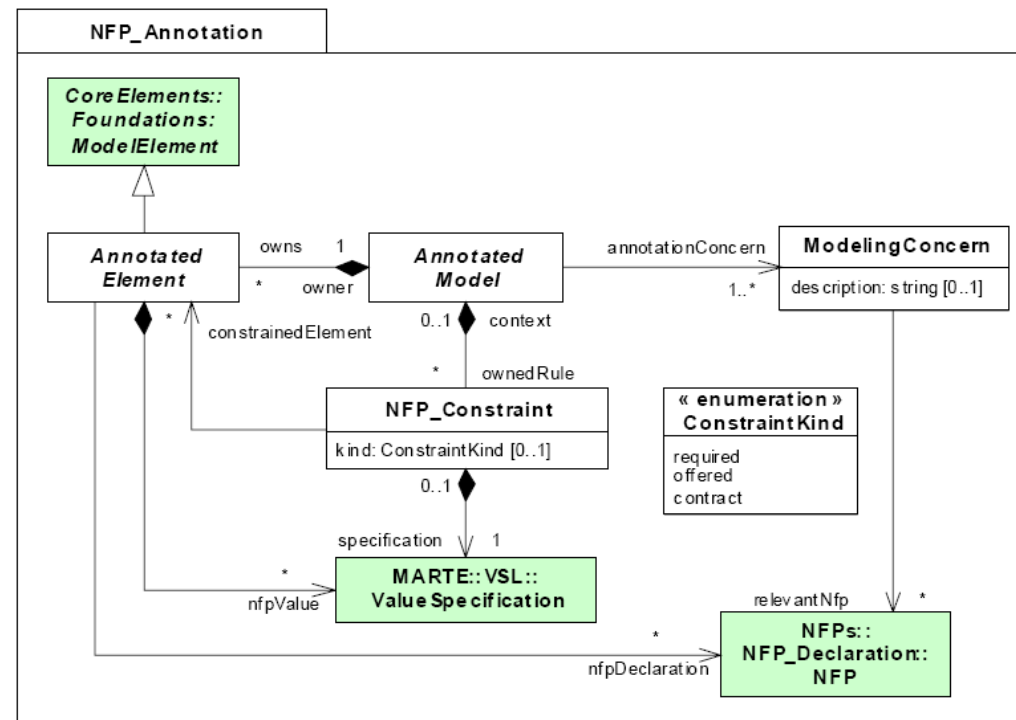
Association to different model elements



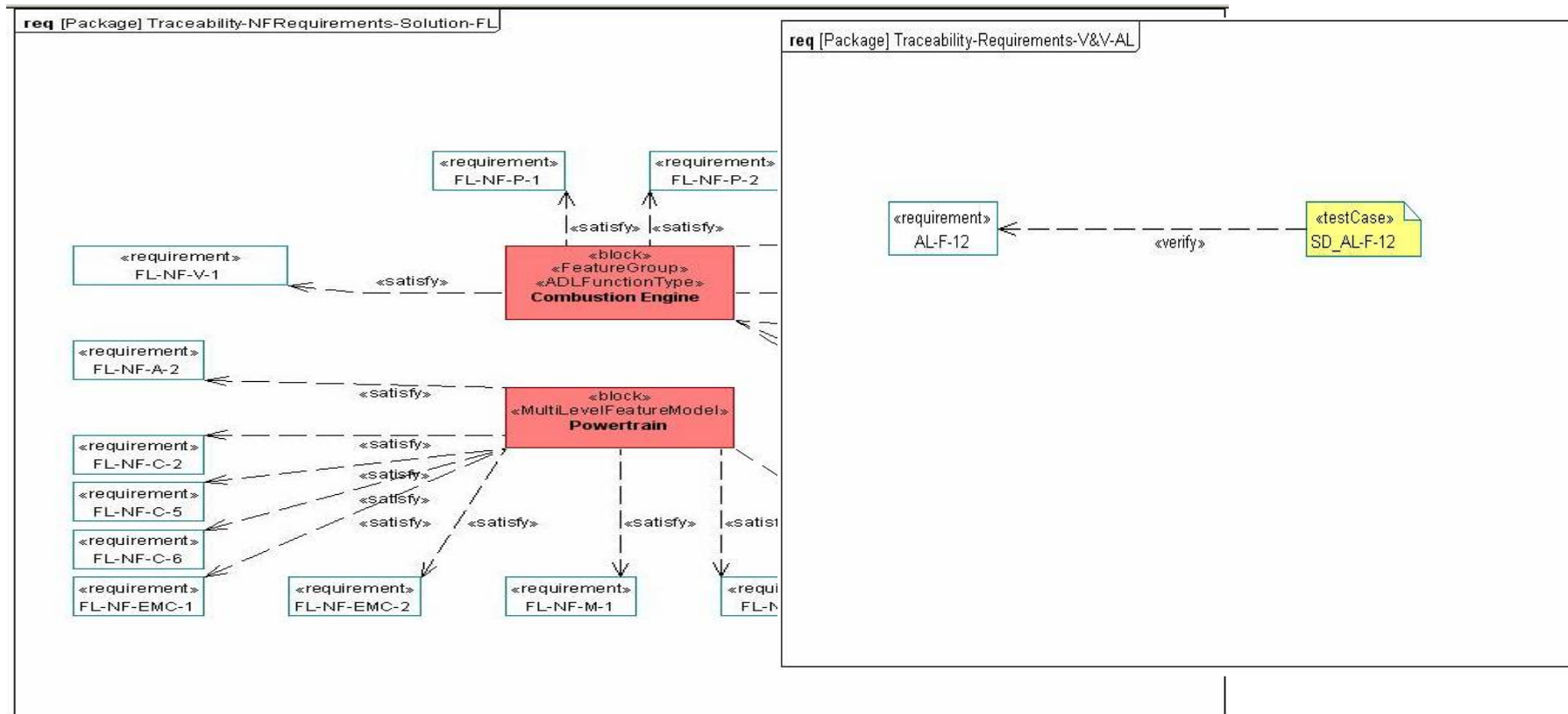
Focus on RT/ES specificities



- A clear connexion between requirements and models
- A specific UML profile for embedded systems: MARTE
- For RT/ES specificities:
 - ▶ real-time requirements ↔ NFP
 - ▶ « Satisfy » relationship usage



- ✓ Requirements ↔ Solutions
- ✓ Requirements ↔ V&V

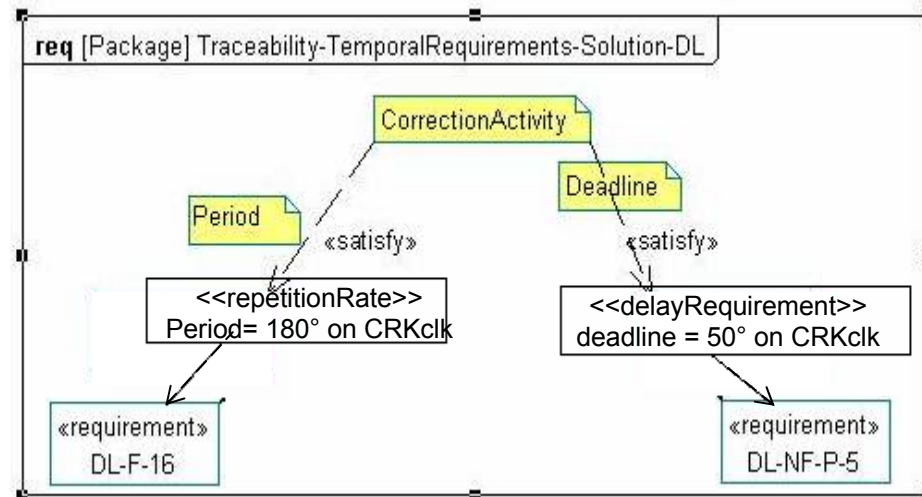
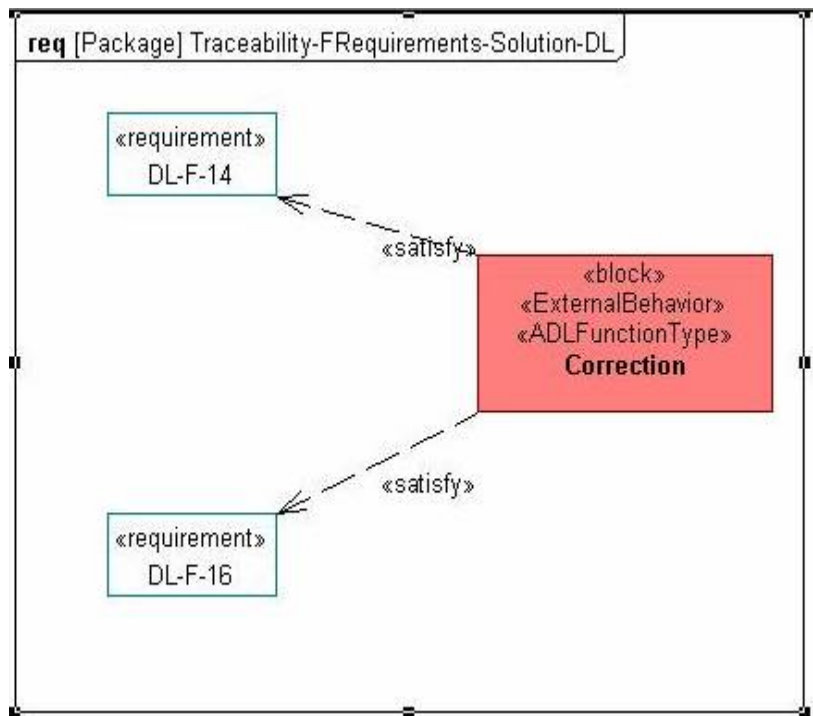


Non-functional requirement consideration – II / II



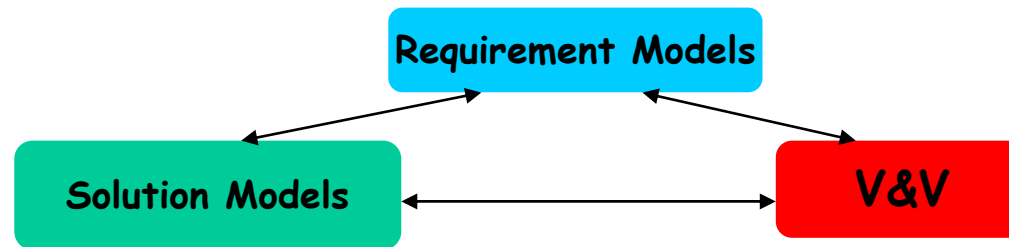
Link between requirements structure and behavioral "timed" models:

- ✓ Functional and delay requirements linked to structure (ADLFunctionType)
- ✓ Temporal requirements satisfy "timed behaviors" of ADLFunctionType

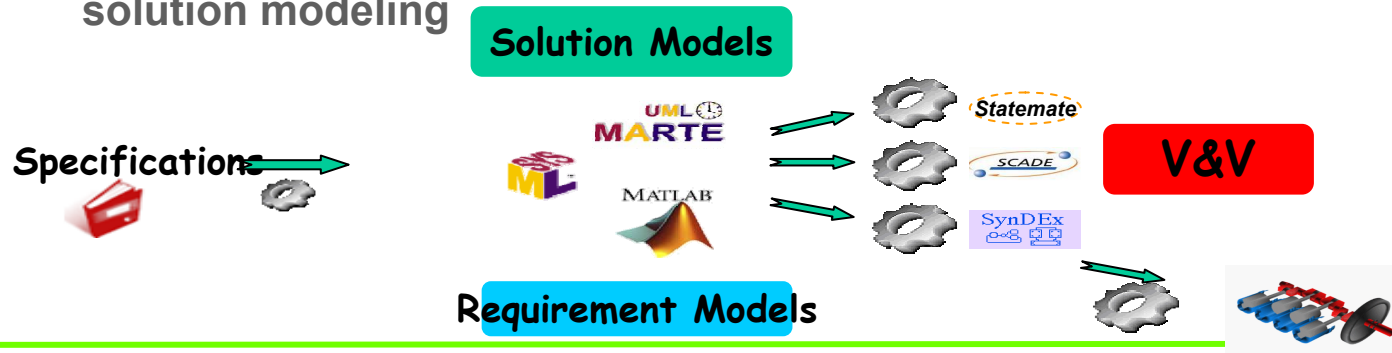


- **Verification of the models properties:**
 - ▶ Based on MARTE profile for RT specificities
 - ▶ For instance: schedulability analysis with SymTA/S tool
 - ▶ We start from a SAM model (*Schedulability Analysis Model of MARTE*).
 - ▶ An automatic transformation to derive scenario-based behavior models and annotations.
 - ▶ Addressed by SymTA/S to check the correct construction of the input model.
 - ▶ **traceability information is essential to facilitate the treatment of the verification result:**
 - ▶ A feedback to the SAM that follows what is correctly or incorrectly checked.
 - ▶ **Association to the initial requirement with respect to the requirement traceability in the models**

- Application in different application domains: automotive, transport.
- Usage in the MeMVaTEx project: automotive context
 - ▶ See <http://www.memvatex.org>
 - ▶ Requirement management
 - Requirement expression
 - & Requirement traceability
 - Usage of a specific profile obtained from the Requirement meta-model
 - ▶ Our solution: a triptych vision towards a separation of concerns but in the UML-based formalism



- ▶ A process that can be connected to heterogeneous formalisms for solution modeling





Conclusion & perspectives



- Definition of a meta-model to deal with requirements and their traceability in an heterogeneous context
- A modeling process connected to a requirement definition process to help in solution modeling with respect to initial requirements
- A traceability support into the models
- A methodological approach for RT/ES systems (automotive for instance)
- A toolled methodology: the Papyrus Eclipse environnement
 - ▶ See Papyrus tool: <http://www.papyrusuml.org>

- Perspectives:
 - ▶ Better consideration of safety & security concerns
 - ▶ A connexion to different analysis tools for MARTE usage and traceability
 - ▶ Traceability concerns during model transformations
 - ▶ Connexion to requirement tools with respect to RIF format
 - ▶ MDE methodology versus verification process
 - ▶ Tool & method improvements to avoid usage complexity

