A model for requirements traceability in an heterogeneous model-based design process. Application to automotive embedded systems

Hubert Dubois, Marie-Agnès Peraldi-Frati, Fadoi Lakhal

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A model for requirements traceability in an heterogeneous model-based design process.

Application to automotive embedded systems

Hubert Dubois - Marie-Agnès Peraldi-Frati - Fadoi Lakhal

N° 7292

May 2010

Thème COM

ISSN 0249-6399

A model for requirements traceability in an heterogeneous model-based design process.

Application to automotive embedded systems

Hubert Dubois[1], Marie-Agnès Peraldi-Frati[2] , Fadoi Lakhal[3]

Thème COM – Systèmes communicants
Projet Aoste

Research report n°7292 – May 2010 - 19 pages

Abstract: Requirements traceability modeling is a key issue in real-time embedded design process. In such systems, requirements are of different nature (software-related, system-related, functional and non functional) and must be traced through a multi level design flow which integrates multiple and heterogeneous models. Validation and Verification (V&V) activities must be performed on models and on the final product to check if they are matching the initial requirements. Results of a design and of V&V activities must impact the traceability information. We propose the DARWIN4REQ metamodel for requirement traceability based on three independent flows (requirement model, solution model and V&V model). The DARWIN4REQ metamodel establishes the link between these flows and allows a full traceability of requirements including the heterogeneous models. This paper presents the DARWIN4REQ metamodel and its use in the context of heterogeneous models for requirement modeling, design and V&V. An automotive application illustrates the approach with SYSML, EAST_ADL2 and MARTE for the design and SIMULINK, SyNDEx and TIMESQUARE for V&V activities.

Keywords: Traceability model, model-based design, validation and verification, real-time embedded systems, automotive application

[1] *CEA LIST DRT / LIST / LISE Gif-sur-Yvette, France* – Hubert.Dubois@cea.fr
[2] *I3S Laboratory, CNRS-UNS-INRIA, Sophia-Antipolis, France*– map@unice.fr
[3] *CEA LIST DRT / LIST / LISE Gif-sur-Yvette, France* – Fadoi.Lakhal@cea.fr

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Un modèle pour la traçabilité des exigences dans un processus de conception basé sur des modèles hétérogènes.

Application aux systèmes embarqués automobiles

Résumé: La traçabilité des exigences est un problème crucial lors de la conception de systèmes embarqués. Les exigences sont de différentes natures (liées au logiciel, au système, fonctionnelles ou non fonctionnelles) et leur traçabilité doit être assurée dans un processus de développement généralement multi-niveaux et qui intègre des modèles hétérogènes. Des activités de validation et de vérification (V&V) sont appliquées aux différents modèles ainsi qu'au produit final afin de vérifier s'ils sont conformes aux exigences initiales. Les phases de conception du système ainsi que les résultats obtenus lors des activités de V&V doivent impacter les informations liées à la traçabilité. Nous proposons un méta modèle DARWIN4REQ pour l'expression de cette traçabilité qui est basée sur trois flots indépendants (modèle d'exigence, modèle solution, modèle V&V). Le méta modèle DARWIN4REQ établit le lien entre ces flots et permet une traçabilité complète depuis les exigences jusqu'aux modèles hétérogènes. Cet article présente le méta modèle DARWIN4REQ et son utilisation dans un contexte de modélisation hétérogènes pour les exigences, la conception et la V&V. Un exemple issu du domaine automobile illustre cette approche avec SYSML, EAST_ADL2 et MARTE pour la conception et SIMULINK, SyNDEx et TIMESQUARE pour les activités de V&V.

Mots clés: Modèle de traçabilité, conception basé modèle, validation et vérification, systèmes embarqués temps réel, application automobile

1 Introduction

The ever increasing complexity of real-time embedded systems raises multiple problems such as the completeness, the consistency, the non ambiguity and the correctness of a design with respect to the initial requirements. In automotive and avionics, the criticism of application imposes for safety critical applications, a full traceability and a verification and validation of requirements (cf. certification standards [19] such as the ISO 61508 standard and the ISO 26262 -automotive domain- or the DO-178B -in aeronautic).

In automotive, the EAST_ADL2 [1] language and the AUTOSAR [2] standard propose a design flow that can be decomposed into several abstraction levels corresponding to the stakeholders view, control engineers view, software engineers view, integrators view. This flow integrates multiple tools and heterogeneous models that capture either functional or non functional requirements (also called extra-functional requirements such as real-time properties, hardware characteristics, performance objectives, variability aspects, safety constraints…). Indeed, depending on the abstraction level, specific formalisms and models are used. Validation and verification activities take part of this flow to ensure the correctness of the design with respect to the initial needs. These activities could be based on simulation, formal verification or test and they must be done at different levels of the design i.e. on the different models or on the final product.

Traceability for either functional or non functional requirements must be maintained through all the levels of a design process, i.e. from the initial problem as expressed by a stakeholder up to the analysis, design, implementation and testing and/or analysis of the final product. This traceability is essential for verification purposes since verification must be associated to the initial requirements, even in the final phase of the software development or verification. In this context, ensuring traceability from initial requirements up to heterogeneous model elements and validation verdicts become a tricky job. This traceability has to consider the different manipulated artifacts initially the requirements, but, also, the proposed solution and the verification and validation artifacts of the process.

Most of existing traceability techniques do not cover all these needs and give partial solutions. In particular, some ones focus on functional requirements and they do not integrate the heterogeneity of the artifacts involved in a design process. Usually, traceability techniques for requirements do not cover the verification and validation steps of the process.

This paper proposes a metamodel for traceability called DARWIN4REQ for a full traceability of requirements from the initial needs through a design process that integrates heterogeneous models and tools for automotive systems. This model establishes the link between three independent flows for requirement modeling, solution design and validation & verification activities. Interactions between these flows are formalized in a traceability meta-model which integrates the heterogeneity of models and maintains the necessary separation of concerns between these three activities. An illustration is presented that covers real-time requirements modeling, their traceability through a design flow based on EAST_ADL2, SYSML,[3] MARTE [4] and validation activities with SYNDEX [5], SIMULINK [6] and TIMESQUARE [13].

The first part describes the needs in traceability management for critical real-time systems. The second part presents the underlying concepts of the DARWIN4REQ metamodel. The third part describes the design methodology we have adopted. The DARWIN4REQ metamodel is presented in the fourth section. The exploitation of the traceability is explained in section V. The section VI illustrates the approach on an automotive example with a special focus on temporal requirements. The last part gives a conclusion and future plans for this work.

2 Concepts underlying traceability

Different surveys [7][8] have shown that there is no standardized definition for traceability. In [7] authors define the requirement traceability as "*…the ability to describe and follow the life of a requirement, in both a forwards and backwards direction, i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of*

on-going refinement and iteration in any of these phases". Three criteria have been identified in [8] to clarify the underlying concepts and classify the area of applicability of traceability methods and associated techniques. The Scope gives the boundaries of traceability (for software and/or system requirements). The Coverage determines the depth of traceability and indicates if the origin requirements are considered or what types of requirements are traced (functional, non functional) and if there exist links with others artifacts than requirements and in-between these artifacts. The last criterion is Analysis provided to establish the safety and reliability of a system. For complex systems, and, more specifically for real-time critical ones, it is essential for traceability techniques to integrate all these criteria. None of the 17 techniques evaluated in [8]fully qualify them.

Additional constraints have been excerpt in [9][10] that concern the backward and forward traceability of requirements throughout the development process. Many changes must be made during the development and the maintenance of a system so, all system components throughout each levels of the development process must be able to be linked back to the requirements. To do that, traceability links must be bidirectional and navigable.

About the exploitation of these links, authors in [10][11] demonstrate that traceability can be exploited differently depending of if it is a simple measure of integration of requirements in the design or if it is a key feature of a quality system engineering process. Backward and forward traceability is also essential when the designer want to study the impact analysis of a modification of a requirement.

In [7], the terms "pre-requirements specification" (pre-RS) traceability, as well as, "post-requirements specification" (post-RS) traceability are introduced. Pre-RS traceability describes the tracing of requirements during the process of requirements investigation. In other words, it is the process of requirements production and refinement. This involves the definition of links between requirements, links between requirements and responsible stakeholders, rationales and sources it originates from, as well as, the tracing of requirements changes. Post-RS on the other hand describes the traceability of these requirements throughout the system to be designed, through its succession of components and tests that verify them. The latter shall support specific analysis tasks like change impact analysis or completeness and consistency checks.

Such traceability has been implemented in a UML model-based approach within the SysML profile. Within this profile, UML has been extended with Requirement Diagram. A requirement is composed of a textual description of the need and a unique identifier. Requirements may be linked to other UML model elements by using the traceability links defines in SysML. Those traceability links requirements (for the decomposition, the derivation and the copy), or links requirements and other modeling elements (for the satisfaction, the verification, the refinement). SysML gives a light description and semantics of these traceability links. In the automotive domain, east_adl2 proposes some clue to consider requirements, but this approach is mainly based on the SysML proposition but with the same drawbacks. .

In our methodology we cover the different topics underlined in the state of the art in order to fully address the traceability of requirements. We also want to cover the system heterogeneity in order to connect the traceability to model-verification and validation purposes.

3 Design Methodology

The traceability model is the cornerstone of a global methodology [12] that aims at considering requirements expression, their evolution during the system development and their validation. As shown on Fig.1, the methodology is a three dimensional triptych composed of three activities: the requirement management, the solution definition and the V&V usage. This approach is justified by the necessary separation of concerns that is required when a certifying process has to be used to certify a product. Each vertex is a multi level model-based flow. The DARWIN4Req traceability model is the central part of this architecture by interconnecting these three independent flows.

3.1 The Triptich description

Fig. 1 represents the triptych and the models involved in the different flows. We can see in the left-hand side the requirement model which is build with initial requirements obtained from requirement tools such that DOORS. The Solution model adopts the decomposition into levels

and the structural behavioral description of EAST_ADL2 AUTOSAR and MARTE**.** In the right hand side of Fig.1, verification and validation is a process that could be connected to solution models and which purpose is to verify and validate the proposed solution and the intermediate models with respect to the requirement model. The verification & Validation flow integrates models such as SIMULINK, SyNDEx and TIMESQUARE [13] for testing and validation activities. As introduced in the first section, we can see that the heterogeneity is a crucial point to consider.

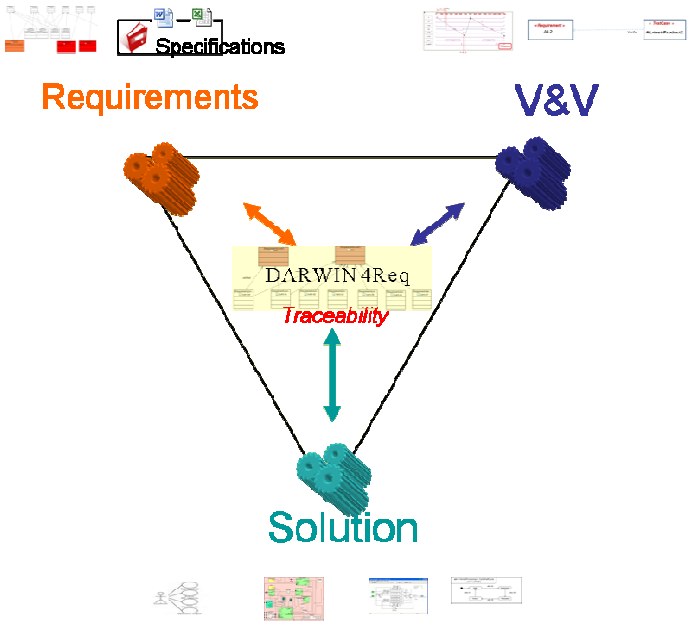

Figure 1: An heterogeneous model-based approach

The traceability model called DARWIN4Req plays a central role in the methodology by connecting requirements to the model-based solution design and to the verification and validation (V&V) artifacts.

3.2 Objectives of the three flows

The requirement model (RM) is a model-view of the initial requirements resulting from a requirement engineering process. Such a representation is essential for designers because the needs become easily understandable, adapted and directly connectable to the models they build. Traceability is also possible between requirement, models and verification and validation artifacts. A clear classification of requirements among functional, performance, safety, variability gives valuable indications for safety analysis and software V&V processes. These indications are inputs for improving the traceability by giving some indications on the artifacts in the solution model side that take into account a requirement. The extension of SysML for classification is one contribution of the proposed methodology. The solution model (SM) refers to models that designers elaborate to satisfy the set of requirements of the Requirement Model. Models can be either UML models or some extensions of UML such as SysML or MARTE. These profiles are used to model the component–based structure or real-time characteristics of the system.
Depending of the application domain, one could identify several languages that may cohabit to design the solution. For automotive applications, the east_adl2 and the AUTOSAR languages are

commonly used, as well as MATLAB/SIMULINK models. In this paper, we considered UML, SysML, east_adl2 and AUTOSAR for designing the solution models as those standards are often used in automotive application domain. Tracing requirements of the RM through the SM gives mandatory for the traceability model to integrate these heterogeneous models.
The V&V model (VVM) contains the models and techniques used to verify and/or test that the Solution Model ensures and guaranties the requirements and that those ones are correctly fulfilled. Verification must be done in both directions, so the traceability links could be navigability in both directions. The heterogeneity of V&V models has also to be considered since several formalisms may be used depending of the property that has to be checked. We consider MATLAB/SIMULINK for testing activities and the SyNDEx analysis tool for schedulabillity analysis is also considered. In addition, V&V is performed concurrently to the solution model design phase.
In that context, the central traceability model has to deal with the specificities of the three models by specifying the connections between the requirements and the different model elements previously presented. Heterogeneity of the models is an essential need of the traceability definition and has to be here considered. We also consider how to reference the results of the requirements satisfaction and verification to fulfill this information in the traceability management.
In order to describe what should be contained in the traceability models, we have defined a requirement meta-model called DARWIN4Req (and its corresponding profile). Since SysML is a UML profile defined as an extension of the UML, traceability links only consider UML-based elements and SysML requirements. The proposed traceability model extends the one proposed in SysML by integrating the heterogeneous models used in the SM and the VVM. We propose a metamodel definition that can be implemented in a UML -based approach in order to be adapted for model-heterogeneity consideration. The meta-model specifies at a fine grain level the interactions between the three flows by taking into account the heterogeneity in the solution modeling and in the V&V techniques .The DARWIN4Req meta-model for requirements is presented in the following section.

4 The DARWIN4REQ meta-model

A meta-model defines the semantics of the concepts used in a model in order to express in an unambiguous way the links between the manipulated concepts and to resolve potential conflicts between them.

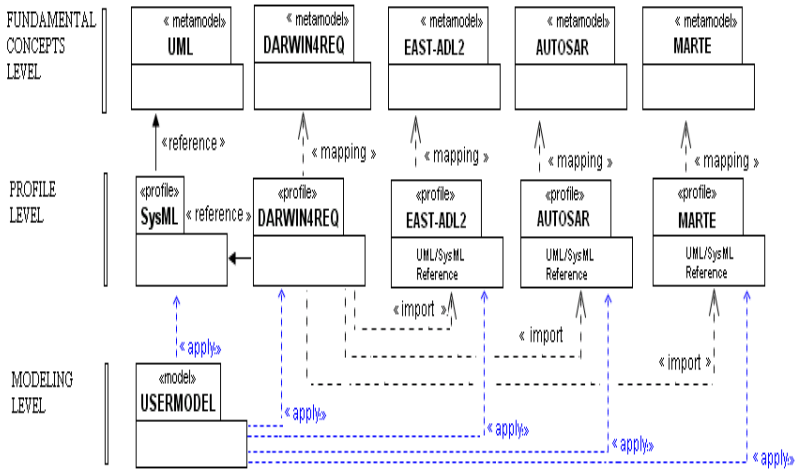

Figure 2: DARWIN4REQ in the modeling process

Fig.2 illustrates the relationships between the different models and metamodels and profiles involved in our approach. On top of Fig 2., we can see the language domain that introduces the fundamental concepts.and the profile level. Metamodels are defined at the first stage of this level. The metamodels considered are: UML, EAST_ADL2 and AUTOSAR. .The DARWIN4Req metamodel defines the fundamental concepts of our approach.
The profile view proposes the languages that will be used for modeling. SysML is a UML-profile and DARWIN4Req is a profile that implements the DARWIN4Req metamodel concepts and that uses the SysML profile. DARWIN4Req profile imports the other profiles defined for east_adl2, AUTOSAR and MARTE as these languages also have their own specific UML-profiles. All these profiles can be used by end-users as it is represented on the lower part of the Fig. 2 in the modeling level by applying the profiles previously declared.
The DARWIN4Req meta-model (cf. Fig..3) is structured in four parts. Apart from a specific RequirementType definition package, the main parts are the requirement definition (RequirementDefinition) package which structures requirement expression according to a classification inspired by [15] the traceability definition (TraceabilityDefinition) package, which details different relationships related to requirement management; the verification and validation definition (VerificationValidation Definition or VVDefinition for short) package, which covers the validation steps of any requirement engineering process and finally the ModelElement package which represents the different heterogeneous model elements considered by the traceability package. The three first packages are described in what follows whereas the last one will be considered in a high-level view.

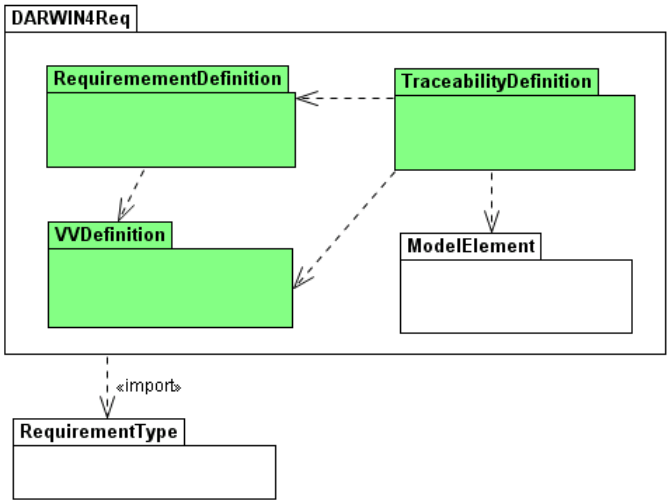

Figure 3: DARWIN4Req meta-model structure

4.1 Requirement Definition

Requirement definition is not fully detailed here. The reader can have a more complete description in [12]of these concepts which are inspired from Hull et al[14]. As the main focus of this paper concerns traceability, of requirements in heterogeneous approaches, we target on the useful properties in this context.Fig. 4 shows the requirement definition part of DARWIN4REQ. The root class `Requirement` stereotyped as <<metaclass>> represents the requirement concepts as described above with the different properties. It encompasses all the properties that characterize a requirement. Thus, a requirement contains, among others, the following properties:

- The `id property`, which corresponds to the requirement's unique identifier;
- The `description` property, used to formulate customer needs in different ways (natural language, drawing, mathematical expression, UML diagram, etc.);

- The classification of the requirement in the development process with the `abstractionLevel` property.
- The `verificationType` property used to describe if the verification used is a model/code review, a test, a proof, etc.
- The `verifyStatus` (resp. the `satisfyStatus`) for memorizing the status of the verification (resp. satisfaction) process. The property value is initiated to `pending` and can be changed into `passed`, `failed` or `inconclusive` depending on the verification (resp. satisfaction) process result.

These two last properties are essential for the traceability analysis as it is explained in paragraph V.

Requirement classification to either functional or non-functional aspects is also an important concern for traceability. We have based our classification on the decomposition proposed by Glinz in [15] and we focus on system requirements.

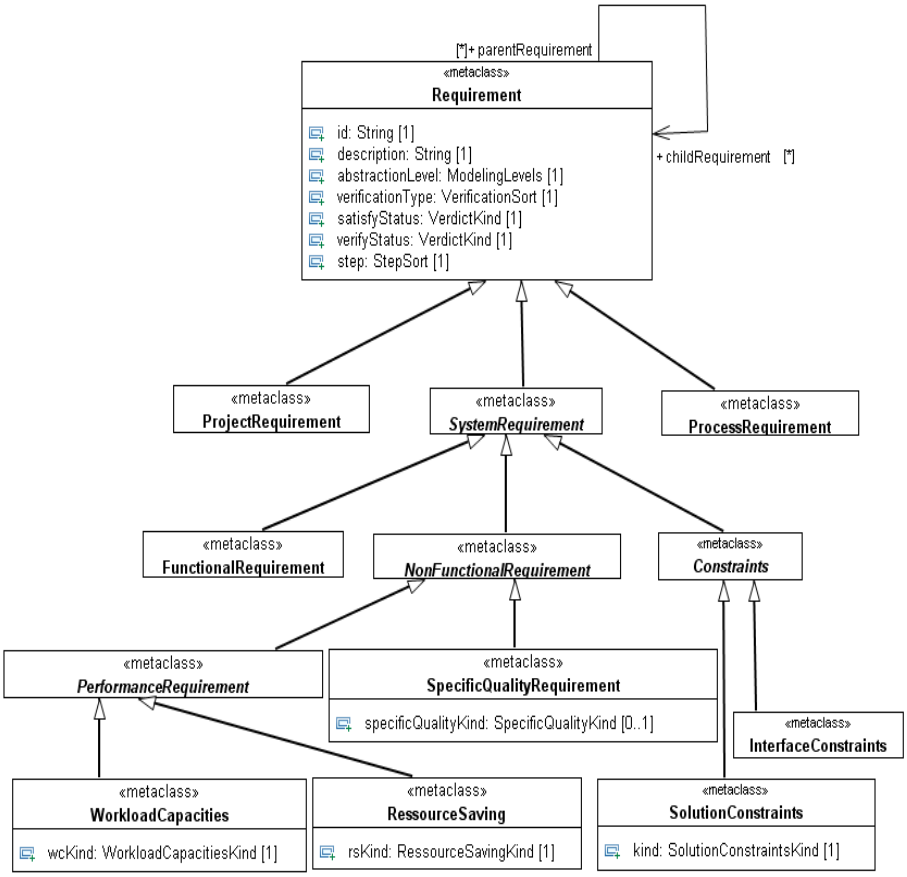

Figure 4:. Requirement Definition in DARWIN4REQ

Fig.4 represents the different requirement concepts described above for classifying the requirements. Some of the meta-classes are abstract (this is the case for the SystemRequirement, the NonFunctional Requirement, the Constraint and the Performance Requirement meta-classes). The difference between an abstract metaclass and conventional meta-classes (also called "concrete meta-class") is that abstract meta-classes cannot be used during the modeling process. They are used specifically to group together several notions and must be further refined by breakdown into submetaclasses. This classification is essential for characterizing the require-

ments and this will help in ensuring a better traceability for specific kinds of requirements (for safety requirement or temporal ones for instance).

4.2 Traceability Definition

Traceability requires first to establish clear relationships between the requirement themselves to handle refinement and decomposition aspects at the different levels of the requirement modeling flow. A second kind of relation must be expressed between requirements and the other artifacts of the solution model and V&V models. Any change, evolution or increment in requirements, in the solution model and in the V&V part must be traced by using the navigability on all the traceability links in both directions. The first subsection explains how requirement traceability is managed when associating one requirement with others. The second one describes the relationships between a requirement and other artifacts of the two other flows (solution models and V&V).

Traceability between requirements

In DARWIN4Req, three relationships are defined to link requirements with one another: derive, decompose and copy. These relations comply with the SysML profile requirement package. The DARWIN4Req meta-model specifies their application. Fig. 5 shows the meta-model view of this package.

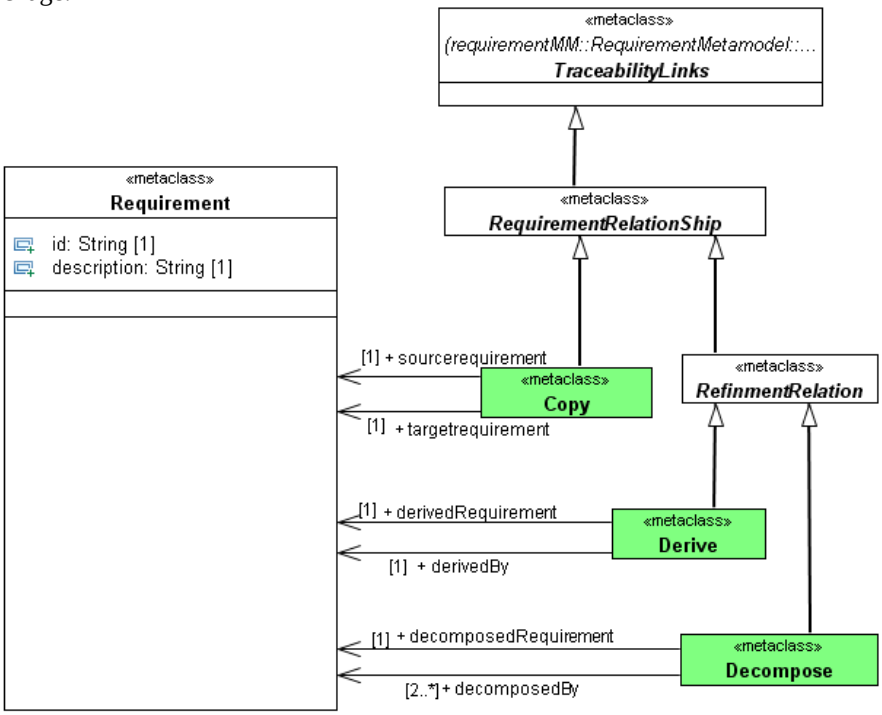

Figure 5:. Metamodel for traceability between requirements.

During the system design process, it may be necessary to modify a requirement (e.g. to add information in order to refine it or to give precision about its content). In this case, a new requirement (called "child requirement") is created to contain this information and a derive relationship is used to connect the child requirement to its source (the parent requirement). A source requirement can be broken down with the derive relationship into one or multiple child requirements.

When a source requirement considers multiple needs (functional and non functional) in a unique description, it is generally recommended to divide the source requirement into as many different requirements as there are distinct sets of information in the source requirement. In this case the

decompose relationship is used to link a source requirement to the different child requirements. The decomposed child requirements can be of different types than the source one. For example a requirement that mixes functional and non-functional information can be broken down into functional and non-functional requirements.
At a given stage of a modeling process, it may be impossible to satisfy (to credit) a requirement because, modeling elements cannot express the requirement needs. For instance, in an EAST_ADL2 decomposition, hardware characteristics are expressed early in the design, but cannot be "creditable" before the design phase which is the third stage of the east_adl2 design process. It is thus necessary to postpone crediting of this specific requirement to a subsequent modeling level, i.e. to descend to whatever the lower modeling level that can credit the requirement. To do so, the copy relationship is used to links identical requirements occurring at different levels. In this case, the requirement description is maintained and only its identifier is changed.

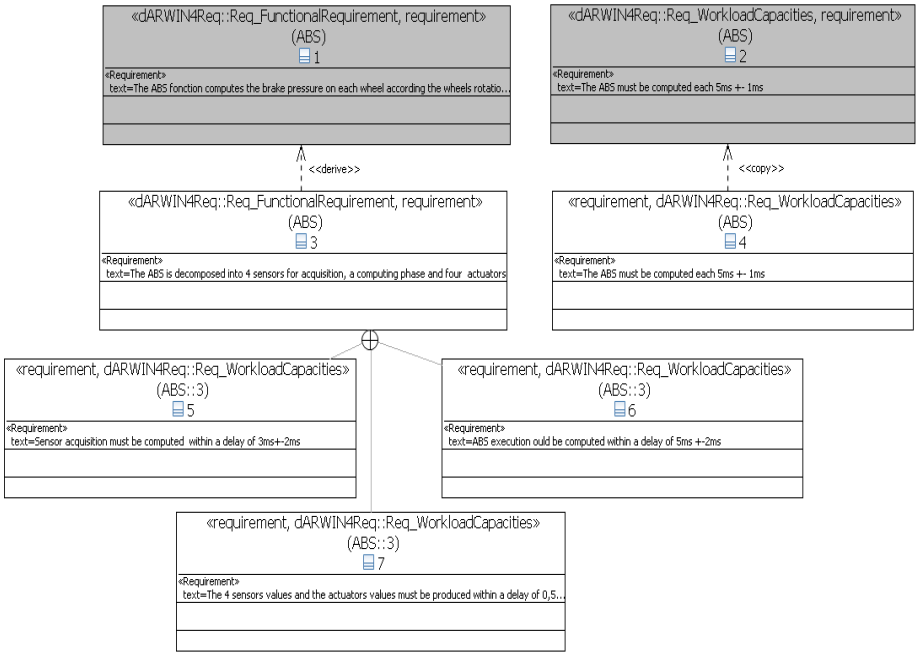

Figure 6: User model with *copy*, *decompose* and *derive* links.

Fig. 6 illustrates a requirement model of an Anti Blocking system (ABS).. Due to limited space, the requirement expression is limited to their description field and their identifier. On the top hand side of Fig.6, there are two requirements from the higher abstraction level, (Vehicule Level). Requirement “1” is refined at the next abstraction level, (Analysis Level) giving rise to requirement named “3”. The requirement “2” cannot be addressed at the Vehicule Level (this is a performance requirement) and, thus, it is copied without any change in the next level as the requirement “4”. The requirement “3” is decomposed into three requirements: “5”, “6” and “7” which consider each one a specific part of their parent requirement.

Traceability between requirements and model elements

Since the third step of a requirement engineering process is system specification, the designer uses the appropriate modeling language and diagram to specify system functions, architecture and interactions according to the expressed requirements. Thus, requirements should be linked with model elements in the solution model as illustrated in the Fig. 7.

To credit this step we essentially use the satisfy relationship. The term “satisfy” means “to conform to a requirement”. The satisfy relationship is set up by the designer during the modeling process. In Fig.7, the satisfy relationship entails identifying in the system model one or several modeling elements that satisfy a requirement. At the contrary, a given model element may satisfy more than one requirement. In both cases the satisfy relationship is used.

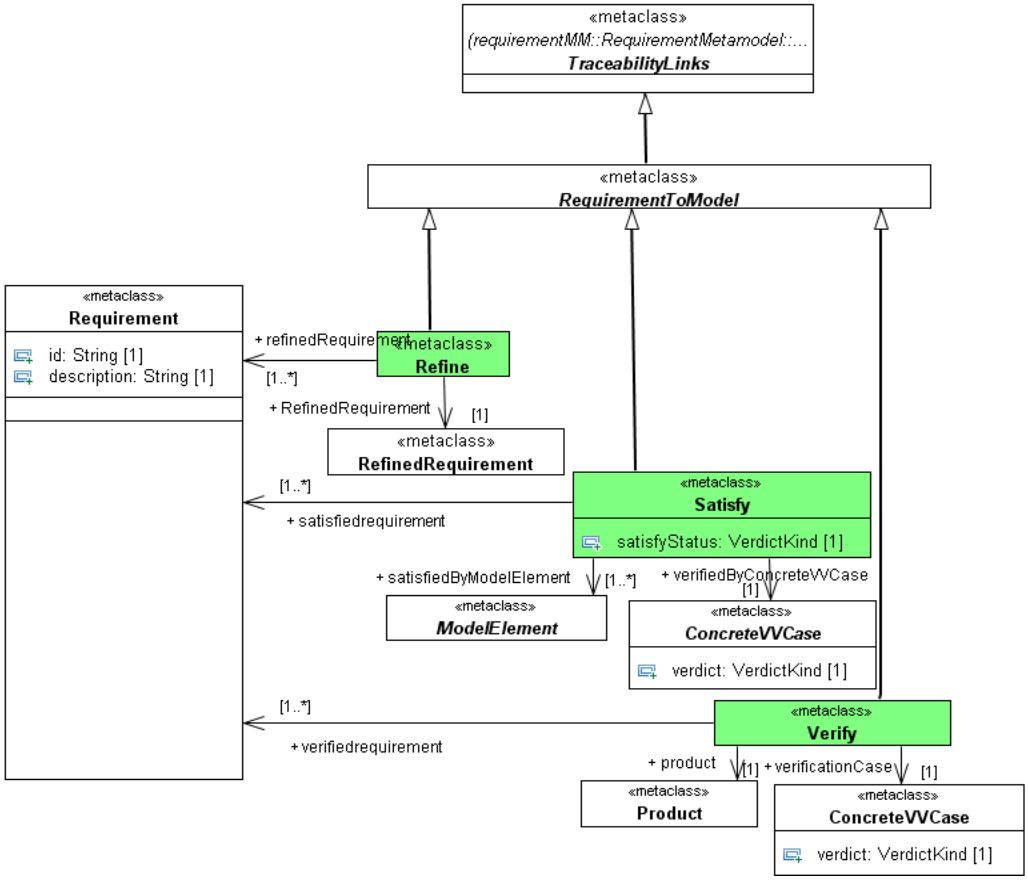

Figure 7: Traceability links with V&V and solution model.

The satisfy relationship is connected to a validation method,(ConcreteVVCase) to confirm by code or model inspection that the requirement is fulfilled by the model. When a satisfaction criterion is met, the property (satisfyStatus) initially set to pending is changed to passed. The ConcreteVVCase participates to the verification and validation model described in the following section.

Another relationship called refine has also been defined. This relationship; not fundamental for the presentation is used when a requirement is refined into a requirement called RefinedRequirement where the description field is replaced by a model element (use case, state machine …). The child requirement called RefinedRequirement is not a classical Requirement since this child requirement cannot be decomposed or copy or derived.

Traceability links with V&V models

The objective of the verification and validation activity is to check that the application behaves according to the specification. We can distinguish three types of verifications: requirement verification, model verification or code (product) verification.

Results of both verification and validation (V&V) are used to demonstrate that a certain system may be considered conform to the initial needs. For this purpose, the V&V models are linked with the requirements. For each requirement an abstract V&V case must be defined. The verify relationship links an abstract V&V with a requirement.
An AbstractVVCase refers to a verification procedure that is based on the requirement specification and textually describes the verification context. It determines which verification method is used and what objectives are set as a function of changes in inputs or variables.

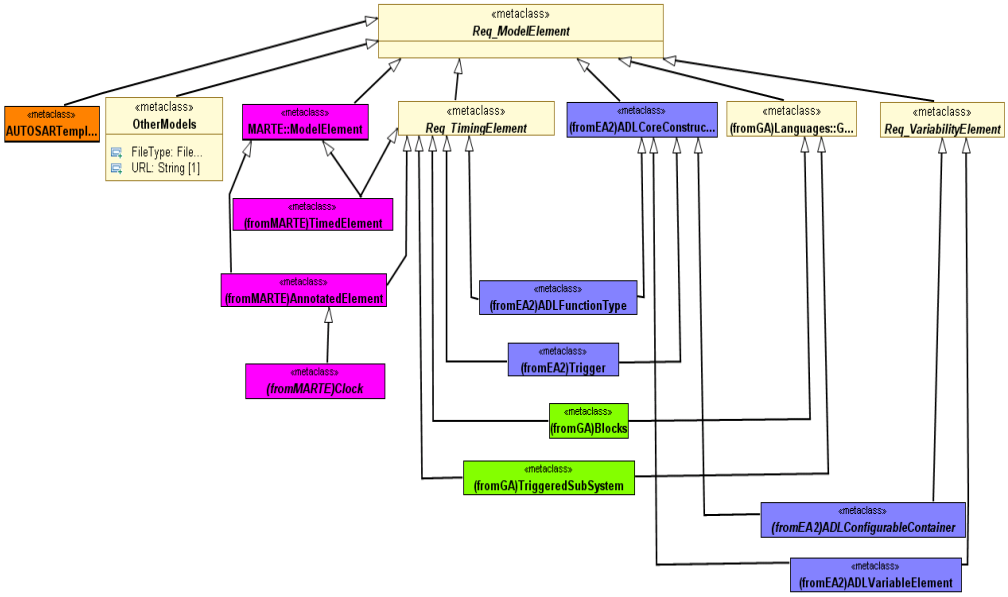

Figure 8: Profile view of heterogeneous models

The verification procedure may start at the first abstraction level to best ensure that all requirements are verified. An AbstractVVCase can be implemented by one or several concrete verification cases i.e: ConcreteVVCase. A ConcreteVVCase may be of different types that correspond to the different possible verification methods, depending on whether the verification concerns the descending or the ascending stage of the V-cycle.
ConcreteVVCase can be of type inspection (code, model or documentation review), or analysis (formal method based on mathematical techniques) or a test case. A test case is a succession of steps that tests final system behavior; the test case should specify the inputs, the precondition used for test performance and the expected outputs. Whatever the type of verification procedure is used, the verdict property of the ConcreteVVCase memorizes the result of the procedure. Possible values for this property are pending, passed, failed, error or inconclusive.

4.3 Management of the heterogeneity in the DARWIN4Req metamodel

The DARWIN4Req metamodel and the related profile are UML compliant, so, they can easily integrate models based on UML (MARTE, SysML AUTOSAR and EAST_ADL2), which participate to the solution model. Moreover, some others such as SIMULINK, SyNDEx and TIMESQUARE may participate to the solution or to the V&V activities. In this context, it is necessary for DARWIN4Req to refer to their metamodel in order to clearly identify in these heterogeneous models, what modeling elements participate to the solution or the V&V and what traceability should be established between them. Thus, at the meta-model level, the modelElement package represents the UML-based and heterogeneous models involved in the approach. The effective import of these models is made at the profile level as illustrated in Fig.8.

5 Usage of traceability links

By using the above mentioned relationships (derive, decompose, satisfy or verify), traceability helps in guaranteeing that the solution models cover all the requirements; and that the model at the different steps of the design process and the final product "correctly" fulfill these requirements. The correctness is established by applying the V&V methods and by reporting these results on verify and satisfy links.

5.1 Coverage of requirements

The requirements coverage can help validate that all requirements are taken into account by the model and by the final product. The requirement coverage may be established by analysing the satisfy links and their status.
In our approach the classification of requirements into abstraction levels, Functional/Non-Functional and types allows a precise view of the coverage. The user may study the coverage with respect to different selection criteria for the requirement: the coverage for temporal requirements, for requirements of a specific abstraction level, etc.
For each requirement in the requirement model, the coverage analysis checks if the requirement is in relation with a solution model element. A total coverage will consist in having a satisfy link from each requirements to a model element.
A complementary analysis based on the satisfyStatus value may add valuable information on the coverage.

5.2 Correctness of a solutions model or a product

Traceability links are also used to help the designer ensuring the correctness of a solution.
These verifications may be done on the solution model or on the final product. In any case, the result is stored in the Verdict property associated to ConcreteVVcase. This property initially set to pending is changed to passed/failed/inconclusive.
If the ConcreteVVcase is associated to a model and a satisfy link, the satisfy status inherit the verdict value. When multiple ConcreteVVcase are associated to a unique satisfy link. The satisfy status is set to passed if all the VVcase are established to be correctly passed.
If the ConcreteVVcase is associated to a product, the verdict of the validation procedure impacts directly the verifyStatus of the requirement. Multiple verifications can be linked to a unique requirement. In this case, the verifyStatus of the requirement depends on all the verification procedure results.
This process can be implemented and we have defined in the Papyrus tool [16] the bases for implementing this process

6 Case study

6.1 General description

The example of an Anti Blocking System illustrates the usage of the traceability model. The ABS architecture consists of four sensors, four actuators and an indicator of the vehicle speed. The sensors measure the rotation speed of the vehicle wheels. The ABS function computes values for the brake pressure to be applied on the actuators connected to the four wheels.

6.2 Initial requirement expression

Stringent timing requirements are imposed on the ABS function such as the latency of sensor sampling (Ls) and the latency of the function execution (Lio). A trigger period for the function is define (R) and an interval delay for inputs and outputs must be respected (Jii Input Synchronization, Joo Output Synchronization).

Table1 : functionnal and temporal requirements

N°	Class	Description
1	VL-F	The ABS function computes the brake pressure on each wheel according the wheels rotation speed values.
2	VL-NF-P	The ABS must be computed each 5ms +- 1ms
3	AL_F	The ABS is decomposed into 4 sensors for acquisition, a computing phase and 4 actuators. .
4	AL-NF-P	The ABS must be computed each 5ms +- 1ms
5	AL-NF-P	Sensor acquisition must be computed within a delay of 3ms+-2ms
6	AL-NF-P	ABS execution should be computed within a delay of 5ms +-2ms
7	AL-NF-P	The 4 sensors values and the actuators values must be produced within a delay of 0,5ms

Table 1, gives some examples of functional and timing requirements imposed on an ABS system. Requirements are identified by a number, the second column provides elements of the classification by indicating the EAST-ADL2 abstraction level (VL, DL, AL …), the type (Functional or Non-Functional) and the classification (Performance, Variability, Safety, etc.).
The initial requirements are expressed and stored in a DOORS database. Requirement expressions comply with the DARWIN4Req model. Fig 6 is the requirement model view of the requirements expressed in Table I and illustrates the traceability links derive, copy, and decompose.

6.3 Traceability links with the solution model

The solution model of the ABS is composed of multiple models (SysML, EAST_ADL2, MARTE). The main relation used to link model element with requirements is the satisfy link. Fig 9 is an illustration of the usage of such link. On the right side three EAST_ADL2 modeling elements satisfy some of the ABS timing requirements. In the same way, others elements that represent the structure and the behavioral parts of the system could be linked with the requirement model.

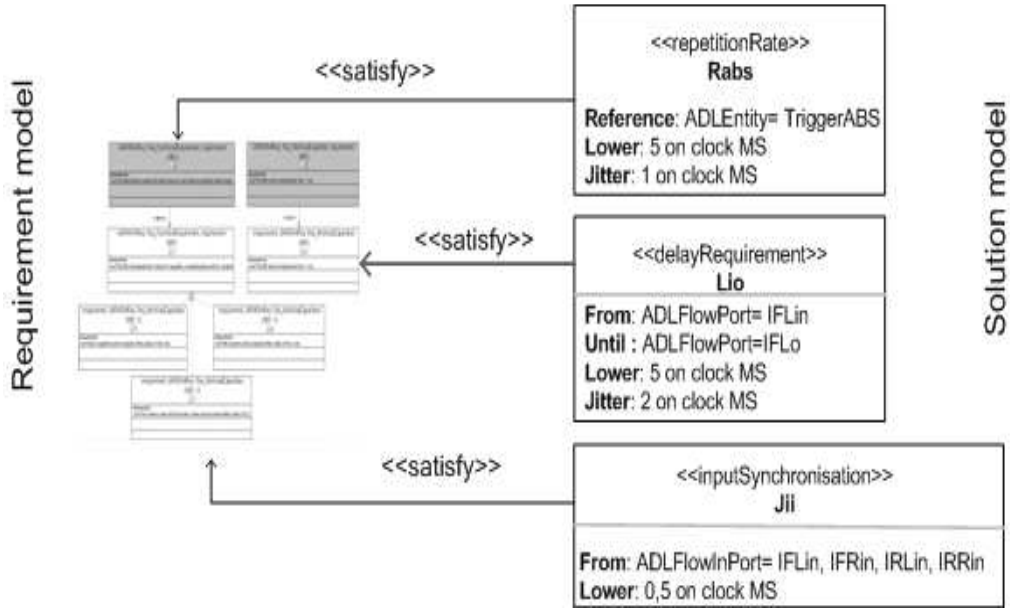

Figure 9: usage of the satisfy link.

This picture is given as a simple illustration. In the uml modeler we used (i.e Papyrus); the requirement model and the solution model are independent packages [16]. The traceability package establishes the links between elements of these two packages by importing their references.

6.4 Verification of temporal characteristics on the ABS

Multiple verifications can be applied on an ABS. We choose to illustrate the verification of temporal characteristics of the ABS.

The requirement model contains four non functional requirements classified as Performance. These requirements concern the AL level. The corresponding abstract and ConcreteV&V cases are represented on Fig.10.

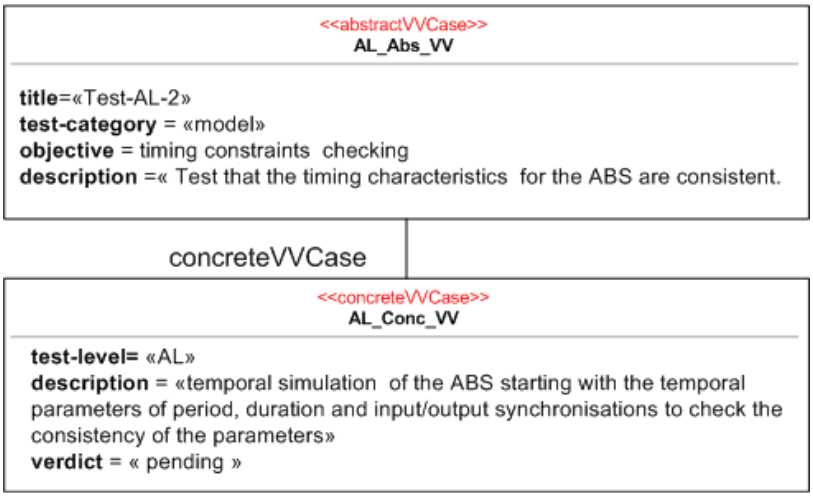

Figure 10: V&V cases for the ABS.

6.5 Feedback on traceability links

Fig; 11 summarizes the verification process and its impact on traceability. A TIMESQUARE simulation model is obtained after a transformation of the timing constraints expressed in the solution model into an executable TIMESQUARE model. Details on the transformation rules are presented in [17]. With the current timing parameters, the TIMESQUARE simulation establishes that the trigger period is not compatible with the sampling period and the function latency values This result is reported to the ConcreteVVCase verdict which pass to failed and the status of all the satisfy links are set to failed too.
A feedback to the requirement model is then necessary to modify the timing parameters in the requirement model and in the solution model.

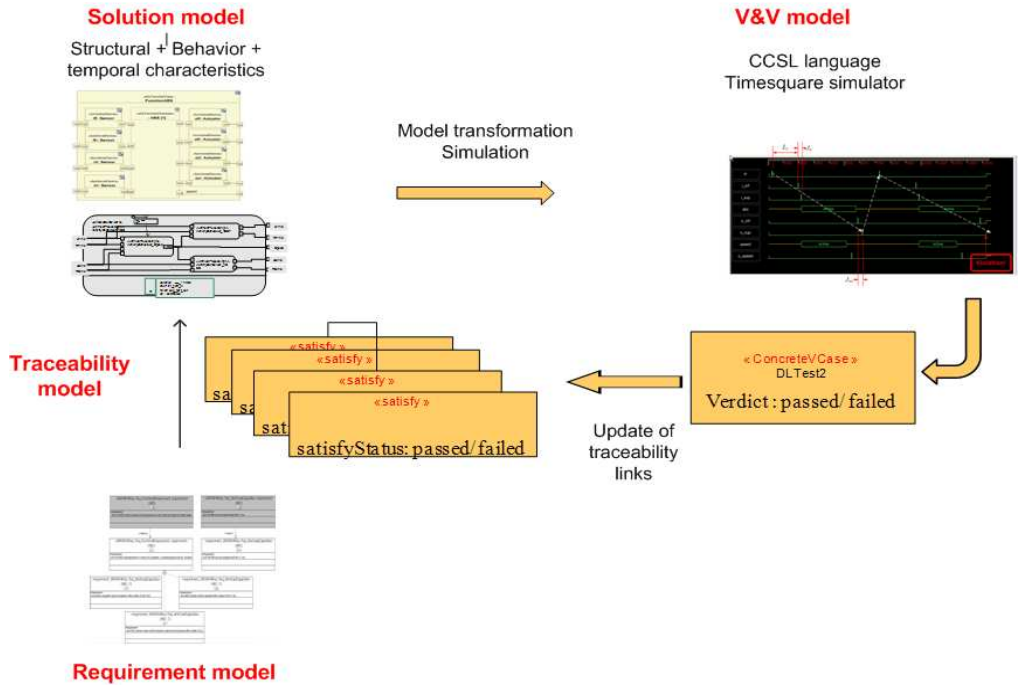

Figure 11: Impact of V&V result on traceability.

7 Conclusion and future works

In this paper, we have proposed the DARWIN4Req metamodel which establishes the traceability links between three distinct flows: the requirement model, the solution model and the verification and validation model for embedded system design.
This approach allows a full traceability of requirements by considering heterogeneous languages for modeling and verifying real-time embedded systems. Indeed, in such systems, heterogeneity is a classical constraint and the different existing approaches generally dedicated to classical software design are not fully adapted. An automotive application illustrates the approach with the use of languages such as SysML, EAST_ADL2 and MARTE for the design and SIMULINK, SyNDEx and TIMESQUARE for V&V activities.
This work has been experimented mainly in automotive applications but the approach is extensible to other application fields. The current European CESAR project4 deals with managing the

[4] See CESAR project webpage: http://www.cesarproject.eu

requirements in a multi-domain perspective and traceability is one of the key-issues of this project.

Acknowledgment

The results presented in this paper have been obtained in the context of the MeMVaTEx project. The authors want to acknowledge the French National Research Agency (ANR) for his financial support.

8 References

[1] EAST-ADL:The EAST-EEA Architecture Description Language, June 2004. ITEA Project Version 1.02.

[2] Autosar.www.autosar.or

[3] OMG. OMG System Modeling Language (SysML) Specification, version 2.1, document formal/2008-11-01, November 2008.

[4] The ProMARTE consortium, UML profile for MARTE, beta 2, ,June 2008, OMG document number : ptc/08-06-08

[5] SYnDEx: www-roc.inria.fr/syndex.

[6] Simulink www.mathworks.com

[7] 0. Gotel and A. Finkelstein, *"An Analysis of the Requirements Traceability Problem"*, Proc. of the IEEE International Conference on Requirements Engineering (ICRE), 1994.

[8] Bashir, M.F.; Qadir, M.A.; "*Traceability Techniques: A Critical Study* ", IEEE Multitopic Int. Conference, 2006. INMIC '06. IEEE 23-24 Dec. 2006 Page(s):265 - 268

[9] Paul Mason, *"On Traceability for Safety Critical Systems Engineering,"* apsec, pp.272-282, 12th Asia-Pacific Software Engineering Conference (APSEC'05), 2005

[10] L. Murray, A.Griffiths, P. Lindsay, P. Strooper, *Requirements Traceability for embedded software- an industry experience report.*.

[11] B. Ramesh, C. Stubbs, T. Powers, and M. Edwards. *Requirements traceability:* Theory and practice. Annals of Software Engineering, 3:397.415, 1997.

[12] A. Albinet, S. Begoc, J.-L. Boulanger, O. Casse, I. Dal, H. Dubois, F. Lakhal, D. Louar, M.-A. Peraldi-Frati, Y. Sorel and Q.-D. Van, *The MeMVaTEx methodology: from requirements to models in automotive application design*, 4th European Congress ERTS (Embedded Real Time Software), Toulouse, France, January 2008.

[13] J. DeAntoni, F. Mallet, C. Andre. *TIMESQUARE : on the formal execution of UML and DSL models*. April 2008, Tool session of the 4th model driven development for distributed real time systems. http://www.mdd4dres.info/_tools/1f3d0539532e6396ad1ecadc4d363a9a

[14] E. Hull, K. Jackson, J. Dick, *Requirements Engineering*, Second Edition, Springer 2005.

[15] M. Glinz, *On Non-Functional Requirements*, 15th IEEE International Requirements Engineering Conference, 2008.

[16] H. Dubois, F. Lakhal and S. Gérard. *The Papyrus Tool as an Eclipse UML2-Modeling Environment for Requirements.* In Proceedings of the Second International Workshop on Managing Requirements Knowledge (MaRK'09), 17th IEEE International Requirement Engineering Conference, Atlanta, USA, September 2009.

[17] F. Mallet, M.-A. Peraldi-Frati and C. Andre, *"Marte CCSL to execute East-ADL Timing Requirements"* IEEE Int. Conf. on Object/Component/Service-oriented Real-Time Distributed Computing, 17-*20 March, 09 Tokyo, Japan , pp249—253*

[18] Michael Edwards, Steven L.Howell , *A methodology for systems requirements specification and traceability*. TR 891-584 Underwater systems department , Naval surface warfare Center., September 1991.

[19] E. Denney and B. Fischer. *Software Certification and Software Certificate Management Systems*. In Proceedings of 2005 ASE Workshop on Software Certificate Management. Long Beach, CA, pp. 1-5, Nov. 2005.