UML-Based Modeling of Non-Functional Requirements in Telecommunication Systems

Mehrdad Saadatmand, Antonio Cicchetti, Mikael Sjödin
Mälardalen Real-Time Research Centre (MRTC)
Mälardalen University, Västerås, Sweden
{mehrdad.saadatmand, antonio.cicchetti, mikael.sjodin}@mdh.se

***Abstract*—Successful design of real-time embedded systems relies heavily on the successful satisfaction of their non-functional requirements. Model-driven engineering is a promising approach for coping with the design complexity of embedded systems. However, when it comes to modeling non-functional requirements and covering specific aspects of different domains and types of embedded systems, general modeling languages for real-time embedded systems may not be able to cover all of these aspects. One solution is to use a combination of modeling languages for modeling different non-functional requirements as is done in the definition of EAST-ADL modeling language for automotive domain. In this paper, we propose a UML-based solution, consisting of different modeling languages, to model non-functional requirements in telecommunication domain, and discuss different challenges and issues in the design of telecommunication systems that are related to these requirements.**

***Index Terms*—Non-functional requirements; Telecommunication domain; UML modeling; Real-Time Embedded Systems; MDA**

I. Introduction

The nature of embedded systems such as resource constraints, close integration and interaction with the environment through sensors and actuators (which can also incur requirements on safety), timing characteristics and lack of traditional user interfaces all bring with themselves requirements that make the design of these systems complicated [1]. Much of this complexity is due to handling a big range of different requirements, solving conflicts and finding the right balance and trade-offs among them. Especially non-functional requirements such as security usually cross cut organizational structures and development teams. Thus traditional functional decompositions do not suit them. However, compared to functional requirements not much work has been done on non-functional requirements and lack of proper methods and techniques for modeling of non-functional requirements and their integration into the development lifecycle are felt [2].

UML profile for Modeling and Analysis of Real-Time Embedded systems (MARTE) [3] is one of the recent and major efforts on modeling Real-Time Embedded Systems (RTES) and the non-functional properties in these systems. MARTE enables detailed modeling of RTES and facilitates their analysis. On the other hand, there is a big variety of systems in RTES domain and to cover the specific aspects and needs of each group of those systems (subdomains), a customized modeling approach is necessary. Such an approach has been used in the automotive domain, leading to the definition of EAST-ADL profile [4] for modeling of vehicular systems.

This paper focuses on telecommunication systems and the aspects that modeling approaches for such systems should be able to cover regarding their non-functional requirements. We propose a UML-profiling approach consisting of features from different modeling languages to answer broader aspects in modeling non-functional requirements of telecommunication systems. One of these aspects is security. We will focus on security in this paper as an example for one of the intrinsic characteristics of telecommunication domain that is also not supported in EAST-ADL. Through an example, we show how it will be possible to model security requirements along with other aspects such as power, in one model while establishing traceability between high requirements and their refinements (lower level ones).

Regardless of the set of non-functional requirements that a subdomain in RTES has, modeling approaches for these systems should provide requirements traceability. This becomes even more important due to limited resources that systems in this domain have; while in other systems, it is usually a lot easier to add extra resources to the system such as additional memory and that way fulfill a requirement. Therefore, a more careful balance and trade-off analysis between requirements is necessary in order to satisfy all of them in RTES domain. Having traceability links among requirements and also between requirements and design artifacts facilitates to perform impact analysis and identify the effects a change on one requirement can have on other parts of the system.

To cover different aspects regarding non-functional requirements in telecommunication systems, we suggest a UML profiling solution consisting of concepts from SysML [5] for traceability, and MARTE for modeling general non-functional properties and their analysis. For security requirements, which are inherent in telecommunication domain but are not covered by MARTE, we adopt from available UML profiles for security, namely UMLsec [6]. Also since MARTE, SysML and UMLsec are UML profiles, they are faster for developers using UML to catch on and they also serve as a possible unifying factor between development departments. A comparison of different ways to define Domain Specific Languages (DSL) and the benefits of each approach are provided in [7], [8]. It is also important to note here that combining different UML profiles is not a trivial task as it may seem and it can incur different problems such as semantic conflicts. These issues are discussed in an interesting work in [9].

The contributions of this paper can be summarized in the following points:

- Showing an approach on how to model non-functional requirements in telecommunication systems
- Identification of issues that should be taken into account in modeling those requirements and the modeling concepts to cover them

As a guideline we use our observations during a project we have done at Ericsson plus the results of other studies such as [2], [10] to describe the modeling challenges. In Section 2, we provide a deeper understanding of telecommunication systems, its characteristics and needs, and the problems observed around non-functional requirements in those systems. We discuss the related work and have a look at some modeling solutions in automotive domain in Section 3. In Section 4, we describe the ingredient concepts of our proposed approach for modeling non-functional requirements in telecommunication domain by highlighting some key relevant concepts offered in SysML, MARTE and UMLsec. Section 5 shows the application of the method as a usage example. In Section 6, we compare the features of our suggested approach with those of EAST-ADL in automotive domain, and finally, in Section 7 we summarize the work and suggest different areas that need to be studied as future work.

II. Motivations

The observations and results in this section are achieved through collaboration with Ericsson engineers (Stockholm, Sweden) and gathered through several meetings with different teams, such as Radio Base Stations (RBS) development group, during CHESS project [11] at Ericsson.

A. Telecommunication Systems

As a type of real-time embedded systems, telecommunication systems have specific characteristics, which incur certain requirements and prioritization of some requirements over the others. These systems need to be secure, are highly distributed, have a dynamic nature, require massive processing capacity and high availability (99.999% availability, which is sometimes referred to as *five nines*), and need to be scalable. The distribution in these systems can be regarded in two perspectives: the distribution inside one node (such as using multicore solutions and distribution of software functions among different processing units) and also the geographical distribution of nodes across different regions and the communication among them.

Typically, telecommunication networks consist of many different types of nodes such as Radio Base Stations (RBS), Radio Network Controllers (RNC), Media Gateways (MGW) and others that span across a big geographical area and communicate over different kinds of lines.

Regardless of the integration and interconnection of different nodes in the network, design of each node is a big complex challenge in itself. For example, an RNC can easily contain between 500 to 700 CPUs, with software functions spanning across several CPUs. This number, however, is decreasing as new processors with higher capacities are produced. This reduction is important for the total cost, power consumption and heat generation of systems. As for functionality and services, in a typical telecommunication system a big number of connections should be established, routed and managed per second. Besides, cost calculation should also be done on them. Moreover, a typical telecommunication system can have a life span of about 20-30 years. Thus upgrade-ability and maintenance of such systems is also of great importance. Software upgrade should be done in such a way to have the least effect on the availability of the system. That is why requirements such as hot-swapping and plugging and the ability to perform restarts at different granularity levels (a single board, collection of boards or a complete node) are highly desirable and demanded in this domain.

B. Problems with Non-Functional Requirements

Due to the hierarchical and subsystem structure of telecommunication systems, first overall non-functional requirements are defined on the system and then they should be refined several times and in each step more concrete and design-decision information is added. However, not all requirements get refined, and as discussed in [10], this leads to weak traceability chains. What can happen is that the requirements that are defined on the system model are *consumed* (meaning that they are read and implemented in the system) and no explicit connection between the design artifact and the requirement leading to that design decision gets established. Also for verification, most of the requirements are tested on a reference configuration and then if some requirements are not met, changes are applied on the system model and again a reference configuration is built with the new requirements. Basically, there are two general issues with this current approach:

- Poor support for traceability of requirements to design artifacts
- The feedback loop for analysis of non-functional requirements takes much time and effort and the wish is to be able to perform verification of non-functional requirements at earlier phases

The organization in large companies usually have a hierarchical structure, which suits the actual system hierarchical structure as mentioned above. According to the study done in [2], this organizational structure matches the system structure well, as subsystems tasks and modules are allocated to specific departments and thus is more suitable for functional requirements. However, this is not the case for non-functional requirements. The problem as mentioned in [2] is that the autonomy of departments at the lowest levels of hierarchy makes management of non-functional requirements harder and that the decisions about these requirements should be done at higher levels of hierarchy and aligned and managed from top to down. This problem becomes more obvious with certain types of requirements such as security, usability and user-interface characteristics, which should be aligned in different subsystems. Thus, non-functional requirements can easily cross-cut organizational structure of a company and therefore a methodology that works for functional decomposition may stop to work for non-functional requirements.

In such organizations, it also happens that different teams may have different interpretations and definitions for some non-functional requirements, which can cause problems for communication between the teams. On the other hand, this also means that people from different teams may talk about a specific requirement using different terms. If we can provide a consistent way of modeling non-functional requirements and a mechanism to establish associations between a requirement or a design artifact and its source requirement, such problems can be mitigated and detected more easily. Also as can be implicitly noticed from the discussion in previous section, there are many requirements that have conflict with others, and trade-off decisions to balance them need to be taken. However, these compromisations and decisions, which may be made inside a subgroup, are somewhat unknown to upper levels and are only known by some engineers working in that section. For example, it is quite common that specific tweaking and settings in the code on bandwidth or memory usage are applied to ensure a certain level of balance between performance and maintainability of the system. Such decisions even if documented are hard to follow and track later on, especially for upper levels in the organizational hierarchy. On the other hand, some requirements that are decided on higher levels are lost in the transition to go to development teams in lower levels of hierarchy. This observation is also in alignment and confirmed by the study in [10], which states the problem as non-functional requirements "are not always available when needed". These issues can be alleviated by applying traceability (which can be traversed back and forth) between requirements and using a better form for representation and documentation of non-functional requirements.

III. Related Work

Requirement Modeling: Telecommunication Standardization Sector (ITU-T) have offered several languages for system modeling in telecommunication domain. Each of these languages try to target different aspects and phases in system development. For example, Message Sequence Chart (MSC) is used for modeling asynchronous interaction scenarios. Specification and Description Language (SDL), which has both textual and graphical representations, uses block, process, channel and signal concepts to describe behavior in communicating real-time systems. At higher abstraction layers and for modeling requirements, ITU-T has suggested User Requirements Notation (URN). URN consists of two notations: Goal-oriented Requirement Language (GRL) to model goals and non-functional requirements and Use Case Maps (UCM) to describe functional scenarios. GRL is used to capture informal system goals, specification and rationals. We refer interested readers to ITU-T website [12] for more information on these languages. Some efforts have been done to define these languages as UML profiles such as [13].

As for general UML-based approaches in RTES domain, MARTE with its expressive power and formal semantics enables capturing non-functional requirements in more formal ways and with necessary details for performing analysis earlier in system development phases. For system engineering, modeling general requirements and the relationships among them, SysML offers Requirements model, and semantics and notations for requirements traceability.

Modeling Security Requirements: There have been efforts on modeling and analysis of security aspects using UML to bring them into earlier phases of development . For example, SecureUML [14] focuses on modeling Role-Based Access Control (RBAC) by extending UML as a profile, while AuthUML [15] is a framework for analysis of access control in the specification phase and thus less suited for code generation. UMLsec on the other hand, uses stereotypes and tag values for modeling general security aspects such as secure links, connections, RBAC, secure information exchange, etc. to enable analysis and early automatic verification (which also matches our goal for early analysis of requirements). A comparison between SecureUML and UMLsec for modeling role-based access control is done in [16]. The UMLsec analysis tool suite can help to identify parts of the model that do not match a specified security requirement. This enables to perform a level of security analysis on the model and find inconsistencies before going into implementation phases. As for other works in this area, the study in [17], for example, introduces stereotypes to specify vulnerabilities so that developers can notice them and avoid in implementation. It also claims that these specifications can be used to generate test cases for security. Article [18] tries to merge Mandatory Access Control (MAC) and Discretionary Access Control (DAC) with RBAC. It is a good work for modeling access control aspects, but lacks other security aspects of UMLsec and their analysis. Doan and Demurjian [19], on the other hand, discuss security analysis based on RBAC and MAC in use-case and class diagrams. Houmb and Hansen [20] introduce SecurityAssessmentUML, which is intended to capture and document the results of risk (i.e., vulnarabiltiies, threats, etc.) identification and analysis. Discussion and comparisons of different UML-based security models can be found in the related work sections in [6], [14], [17]–[19].

Requirement Modeling in Automotive Domain: As an example of a UML-based domain-tailored approach, EAST-ADL has been developed in automotive domain for modeling software architecture and electronic parts of a system. By complementing and making use of general available modeling solutions in RTES domain, EAST-ADL tries to cover the specific requirements of automotive domain. It adopts concepts from UML, AADL [21] and SysML to provide modeling semantics aligned with AUTOSAR [22] specification. AUTOSAR focuses on lower design levels such as component model, software modules, control units, APIs and implementation parts of automotive systems.

For modeling requirements, EAST-ADL makes use of SysML requirements semantics and specializes them to match automotive domain (e.g., definition of timing, delay and safety requirements). However, it does not provide enough features to enable some analyses such as scheduling and timing verifications earlier than implementation phase [23]. There are studies such as [23] that suggest decorating EAST-ADL models with some features from MARTE such as timing and allocation packages to enable early scheduling analysis. TIMMO project [24] is one of the efforts using this idea to complement timing model of EAST-ADL for automotive domain. In general, EAST-ADL and its requirement model may not be appropriate and compatible as a whole for requirements in telecommunication domain. It does not cover security aspects, which are important for telecommunication systems, is aligned with EAST-ADL's specific abstraction levels, and is based on concepts like ECU, VehicleFeature, AutosarSystem, and Sensor which are not relevant for telecommunication systems. In order to better capture requirements of telecommunication systems that originate from their specific characteristics such as intensive performance demands, distribution, use of multi-core solutions, virtualization and hierarchical schedulers, etc. a tailored solution for this (sub)domain is required.

IV. Suggested UML profile

Adopting a model-based approach for the development of telecommunication systems helps to raise the abstraction level and cope with the design complexity. This also targets the challenge to shorten the feedback loop and enable analysis in earlier phases of development.

In this section, the key concepts that a desired UML profile for telecommunication systems should be able to offer are discussed. We explain traceability concepts from SysML, modeling general non-functional requirements with MARTE highlighting its relevant and interesting features for telecommunication domain and how to model security aspects along with an example of its analysis. Later in section 6, we compare the features of our suggested UML profile with EAST-ADL.

A. Modeling Traceability Using SysML

For modeling of requirements, SysML provides a specific diagram, which can be a solution to the issues regarding management of non-functional requirements of telecommunication systems identified in previous sections. An important feature of SysML is to represent requirements as first-class model elements. So requirements are included as parts of the system architecture and have semantics [25]. This also enables establishing relationships between requirements and other model elements showing, for example, design artifacts implementing and satisfying a requirement. It is possible to decompose requirements and create a hierarchy of requirements, which is needed to cope with the complexity of requirements faced in telecommunication domain. SysML provides different types of associations among requirements, which include: *copy*, *deriveReqt*, *satisfy*, *verify*, *refine* and *trace*.

The counterpart of these associations are *derivedFrom*, *satisfiedBy*, *refinedBy*, *tracedTo*, *verifiedBy* and *master* properties that a requirement element can have. For example, *satisfiedBy* property of a requirement element contains the information of the model element that satisfies this requirement (counterpart of *satisfy* association). This way, SysML facilitates traversing back and forth between requirements and also model elements from high level departments in organizational hierarchy to lower level departments and development teams.

Another feature that SysML provides is requirements table. Requirements table provides traceability information for requirements in a single view, which is very helpful in managing the big number of versatile requirements in telecommunication systems. In this tabular representation of requirements, information such as requirements properties and types, dependency relationships with other elements/requirements and other information such as design rationale and test procedures may be included. By going through this table, it is possible to analyze the change (e.g., modification, deleting) effect of one requirement on other requirements in the systems. So basically, by providing different types of association and dependency and the tabular representation of requirements, SysML can answer problems identified for traceability and impact analysis of requirements in a complex and hierarchical telecommunication system. Moreover, by using stereotypes it is possible to extend SysML, which makes it very flexible to add new semantics such as new types of associations or requirements. An example of this extension is provided in [25], where three stereotypes for functional requirements, non-functional requirements and external interface are defined and used to model a system.

B. MARTE for Non-functional Requirements and Analysis Support

To represent the properties of non-functional requirements such as timing constraints in a formal way, MARTE provides rich modeling semantics. MARTE profile consists of different subpackages and in this section we try to identify packages and semantics in them, which serve to represent the type of non-functional requirements we identified in a telecommunication system.

MARTE NFP_Types, Value Specification Language (VSL) and the stereotypes defined in NFP package (Non-Functional Properties) help to define different non-functional properties specific to different domains. NFP package makes it possible to define percentage, dimensions, measurement precision and similar concepts for non-functional properties. Examples of basic NFP types already defined in MARTE type library can be power, frequency, duration, energy, weight, length, arrival-pattern (periodic,aperiodic, sporadic), price, etc. For time specifications, MARTE offers the time package and representation of time in MARTE can be in the form of a physical (continuous or discretized) or logical clocks (processor cycles, engine rotation, algorithmic steps…). The concept of multiform time provided in MARTE is very useful for telecommunication domain, which has already started heading for multi- and many-core solutions. The semantics to model the execution platform (operating system, virtual machines, hardware) are packaged in Generic Resource Modeling (GRM), Software Resource Modeling (SRM) and Hardware Resource Modeling (HRM). With SRM it is possible to model concepts such as resources, services, concurrency and mutual exclusion features in a Real-Time Operating System (RTOS) as well as virtual machines, which are used in telecommunication systems. The stereotypes in HRM package enable modeling of processing units, different levels of memory, devices and their physical aspects such as layout in the system, power consumption, and heat dissipation. These concepts can be used to target non-functional requirements such as cost sensitivity, execution capacity, and environmental requirements (layout, size, structure, etc.).

An interesting feature provided in GRM is the modeling of primary and secondary schedulers, which enables modeling of systems having hierarchical schedulers. This is helpful for telecommunication domain in which, use of hypervisors and virtual operating systems on top of another operating system is common.

To model dependability requirements (reliability, availability, maintainability, safety), which is an important feature of telecommunication systems, there is a suggestion for an extension to MARTE, that is introduced in [26] as Dependability and Analysis Modeling (DAM) (sub)profile and offers relevant concepts such as threats (fault, failure, error, hazard, accident), maintenance, redundancy, etc.

C. Covering Security Aspects

Security in embedded systems is becoming more important and gaining greater attention. More mechanical parts are replaced by computer systems and the use of wireless technologies for communication between different units is becoming a dominant trend. In automotive domain for example, features such as traffic and accidents notification systems, built-in bluetooth devices and distance calculator between cars are representatives of such cases that require communication with other cars and devices. Such features along with electronic access controls (e.g., access to the vehicle internal bus and electronic locks) also open up the system for more security threats.

Due to the nature of systems in telecommunication domain, which naturally involve long distance communications and at a big level of distribution and scalability with many nodes and access points on the way, security aspects have always been an unavoidable part. A single telecommunication node such as a Radio Base Station (RBS) can serve different requests from different sources and these operations should be kept separate from each other keeping data intact and safe from interference. It becomes more critical when we add to the picture other services in the system such as call cost calculation for users and (recently) data traffic including images, emails, and other sensitive and personal information. However, in the design of a system, security considerations should not be considered as an add-on, but they should be taken into account from early phases of design.

UMLsec covers a broad scope and has a versatile tool suite for analysis. Using that we can complement modeling of security requirements as first class entities. With the help of SysML, relationships between them and other non-functional requirements and also design artifacts can be added and detailed non-functional properties using MARTE can be specified for them if necessary. So for example, it becomes possible to model nodes in a system as resources using MARTE GRM package, and then define necessary users, roles and communication security requirements between the nodes using UMLsec profile. The relation between these elements and the source requirement element incurring such security design can be established using SysML requirements concepts.

UMLsec offers several stereotypes such as *Internet*, *wire*, *LAN*, *encrypted* for physical links between nodes. These concepts can be applied on communication links in telecommunication systems. Each link type in UMLsec is defined as prone to different types of threats (*read*, *insert*, *delete*) from different attackers. For example, a link stereotyped as *LAN* or *wire*, has no threat from a default (external) attacker. However, an insider attacker can still pose *read*, *insert* and *delete* threats regarding the packets and information transferred on such a link. If it is needed to define new types of links, attackers or threats, UMLsec allows this. The new concepts can be defined in the UMLsec appropriate format in an XML file, so that the analysis tool can perform correct analysis based on these custom concepts on a model that makes use of them. In this sense, the analysis tool is flexible and extensible. *Secrecy* stereotype that is used on dependency relationships applies a secrecy/confidentiality requirement on the elements of dependency base class. This way, we specify that there is a secrecy requirement for the involved elements.

V. Modeling a Security Requirement Using the Suggested Profile

In a typical 3G telecommunication network, different groups of Radio Base Station (RBS), Radio Network Controller (RNC) and Media Gateway (MGW) nodes are connected and communicate. When a Mobile Equipment (ME) wants to join the network, it starts communicating with RBS and authenticating itself to the system. Security operations such as key exchange take place through the communication path from the mobile equipment to the RNC. In our case study, we have two RBS 3202 nodes that communicate with an RNC. The output power requirements for RBS 3202 are as follows:
Req1: Optimized Power 15W, Standard Power 20W, High Power 30W and Dual High Power 60W.
One of the security requirements that exist for the connection between the RBS and RNC is:
Req2: Data communication between RBS nodes and RNC should only be readable by inspection group.

The second requirement incurs that no one from outside and also inside of the network should be able to read the data traffic on the links between RNC and RBS except users in the inspection group. Thus the data should be encrypted using a specific key for this group. We try to violate this in our

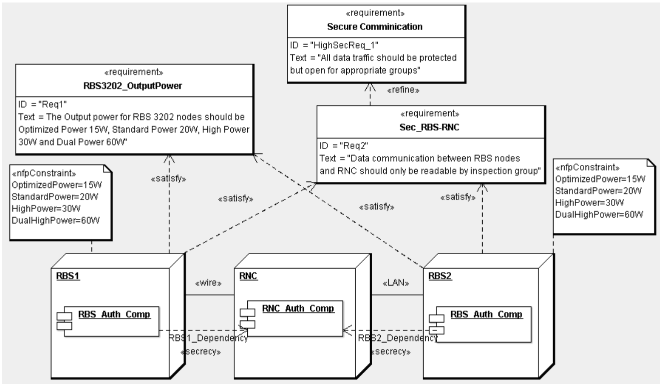

Fig. 1. Security Requirement on RBS Nodes

```
::::::::Against Default Attacker
=======Here begins the verification
The name of the dependency is RBS2_Dependency
The stereotype of the communication link of the dependency RBS2_Dependency is LAN
The stereotype of the dependency is: secrecy
* The UML model satisfies the requirement of the stereotype secure links.
...
::::::::Against Insider Attacker
=======Here begins the verification
The name of the dependency is RBS2_Dependency
The stereotype of the communication link of the dependency RBS2_Dependency is LAN
The stereotype of the dependency is: secrecy
* The UML model violates the requirement of the stereotype secure links, but
it has been fixed.
...
```

Fig. 2. Result from UMLsec Analysis Tool

example model by using unencrypted links and then perform analysis on the model.

As shown in Figure 1, the requirements and the relationships between them and design artifacts are modeled using SysML concepts. MARTE non-functional concepts (i.e., *nfp, nfpconstraint, PowerUnitKind and NFP_Power*) are used for modeling output power requirements of RBS nodes. Security concepts in our model are represented using UMLsec stereotypes. The link between RBS1 and RNC is marked with *wire* stereotype and the one between RBS2 and RNC is marked with *LAN* stereotype (in UMLsec wire and LAN are two different security stereotypes that can incur different security characteristics).

Doing analysis using UMLsec analysis tool on the model yields the result that is shown in Figure 2. The important part in this analysis output (marked with *) is that *LAN* and *wire* links are not readable by a default (external) attacker thus the model satisfies the secrecy requirement for this attacker type, but an insider attacker on LAN or wire can access the information and therefore the model violates the requirement. Although UMLsec has a general *encrypted* stereotype to label encrypted communications, it is also possible to define a custom stereotype for example as "*Uniquely encrypted by SIM ID*" and define different threats that different attackers can pose on these links such that only inspection group users can have access. Then we can use this stereotype on the links instead of *LAN* and *wire* that we used earlier, to create a model that satisfies the requirement and verify it with the analysis tool.

VI. Discussion

As mentioned in the related work section, EAST-ADL is a modeling solution for automotive domain that is built using a similar approach to what we proposed here by adopting from several UML profiles. It is successfully accepted in the automotive domain and its usage together with AUTOSAR is gaining more momentum. In table I, a comparison of capabilities of our suggested solution using MARTE plus SysML and UMLsec against those of EAST-ADL is presented, with a focus on modeling concepts and features that are necessary for NFRs in telecommunication domain (e.g., processing capacity and memory consumption that are important for performance analysis). It summarizes the concepts we discussed and identified in previous sections. The star mark in the table is used to indicate that the feature is not enough/fully supported, such as the dependability modeling in our approach. However, it can be covered by using the DAM profile introduced earlier, which is built as an extension to MARTE. Modeling of time for schedulability analysis support in EAST-ADL needs also to be complemented (as is investigated in [24]).

TABLE I
Comparison of the suggested UML-based modeling solution with EAST-ADL of Automotive domain.

Modeling Feature	Our Approach	EAST-ADL
Generic NFRs (SysML Style)	✓	✓
Traceability of NFRs	✓	✓
Timing, Clock, Schedulability Support	✓	*
Memory consumption	✓	✗
Processing capacity	✓	✗
Power consumption	✓	✗
Virtual machines and hierarchical schedulers	✓	✗
Hardware platform	✓	✓
Multicore	✓	✗
Allocation and Deployment	✓	✓
Communication media	✓	✗
Safety	✗	✓
Security	✓	✗
Variability (product families)	✗	✓
Methodology (e.g., abstraction levels)	✗	✓
Dependability (e.g., fault, error...)	*	✓
Synchronization mechanisms	✓	✗
Arbitrary Non-Functional Properties	✓	✗
Component model	✓	(AUTOSAR)

From the table, it can be seen that by *tailoring* a UML profile for telecommunication systems based on the concepts in the three available profiles we discussed in this paper (MARTE, SysML and UMLsec), it is possible to better cover the requirements of telecommunication systems, than just reusing only EAST-ADL modeling semantics from automotive domain, which are tailored for the needs of systems in that domain. Security is one of the specific needs of telecommunication systems that is not supported by EAST-ADL and has not been in the main focus in automotive domain (so far). While on the other hand, safety requirements, which are very important for automotive systems, are explicitly supported in EAST-ADL. For differences between safety and security requirements, interested readers can refer to [27].

While in this paper, we discussed a UML-based solution by adopting and tailoring already existing profiles, other methods of defining a specific language for modeling telecommunication systems are, of course, possible. However, although designing a domain specific language from scratch may match the needs of telecommunication systems better, it also implies the need to design dedicated modeling tools, and additional costs for training the users to learn the new language. On the other hand, some of the benefits of a solution based on UML are that many users are already familiar with UML, and thus, the learning curve is smaller. Also, there are already many tools for creating UML models which can be used 'out of the box' [7], [8]. One point to remember though is that, as mentioned before, combining different UML profiles can be problematic in some cases. For example, there is FlowPort both in SysML and MARTE. However, the semantics of FlowPort in SysML are different from those of MARTE. A systematic approach is suggested in [9] to ensure consistency in merging UML profiles.

Regarding the management of models, based on the features of the modeling tool, there can be several scenarios. For example, different models can be created for different aspects of the system. This can also help with the analysis, as one model for each type of analysis can be created. However, maintaining consistency between different models of the system and redundant information modeling are some of the challenges of this approach. Another scenario could be to have one single model for the system, and then have the modeling tool provide different views of the core model. This way, a user can just focus on the aspects of his/her interest in each view, while modifications are persisted into one single model representing the system. This method is under development in CHESS project [11].

As for the analysis of the models, although this topic is not the main focus of this paper, but we provide some hints here. Basically, the process of analysis can be different for various analysis tools, and depending on which types of analysis are of interest for different end-users. In case of having just one single model of the system, if an analysis tool can ignore non-relevant model elements and perform analysis on the relevant parts, the model can be fed as input to the analysis tool directly. However, if non-relevant model elements may cause problems for the analysis, then it is possible to use model transformation to extract only the relevant ones into a new model appropriate as input for the analysis tool. Also, if the input model of any analysis tool has its own specific meta-model, then model transformation techniques can again be used to transform the original model into a new model conforming to the meta-model of the analysis tool.

VII. Conclusion and Future Work

In this paper, we discussed several challenges in modeling non-functional requirements in telecommunication domain. We also suggested a modeling approach for representation of non-functional requirements and their properties in this domain. Our approach was to consider telecommunication systems as a subdomain of RTES and therefore adopt from available modeling solutions for non-functional requirements and their analysis that already exist in RTES domain. Some concepts of MARTE that can cover the requirements of telecommunication systems were highlighted. For traceability aspects, SysML and the features it provides in establishing traceability in modeling of non-functional requirements were introduced. Finally, as a specific and intrinsic requirement in telecommunication domain, it was shown how it is possible to model and analyze security that is addressed in our suggested approach by adopting UMLsec. This way, we showed not only how it is possible to model different types of non-functional requirements, but also how model-based analysis can help with the need to perform analysis of non-functional requirements at earlier phases of development and therefore reduce time and cost. In CHESS European project [11], we are developing a similar solution by using subsets from MARTE, SysML and DAM profile (without security considerations yet) to generate code for telecommunication systems (in this case, Ericsson platforms) considering and preserving non-functional requirements modeled using the mentioned subsets.

As further studies, it is necessary to augment the suggested approach in this paper, such as introducing it as part of a well-structured methodology similar to the methodology suggested in [28]. This methodology is more suited for automotive domain as it makes use of EAST-ADL and its abstraction levels. Applicability of the same concepts to telecommunication domain could be an interesting topic to investigate. Especially that EAST-ADL offers concepts for modeling *variability* requirements, which can be very useful in telecommunication domain for modeling product families, targeting cost-sensitivity non-functional requirements and performing cost analysis.

Also other challenges that exist regarding non-functional requirements in a model-based development approach can be guaranteeing and preservation of these requirements on the target platform, introducing runtime adaptability and re-configuration based on the requirements and handling their violations.

As a last note, in this paper we set the basis for a UML-based solution for telecommunication systems similar to EAST-ADL in automotive domain. While it was demonstrated how we can relate high-level and abstract representation of an NFR such as security with its lower level realizations and perform security analysis on it, a full scale solution needs contributions from different industrial partners active in the domain as has been done in the process of defining EAST-ADL and AUTOSAR.

References

[1] T. Henzinger and J. Sifakis, "The embedded systems design challenge," in *Proceedings of the 14th International Symposium on Formal Methods (FM), Lecture Notes in Computer Science*, August 2006.

[2] A. Borg, A. Yong, P. Carlshamre, and K. Sandahl, "The bad conscience of requirements engineering : An investigation in real-world treatment of non-functional requirements," in *Third Conference on Software Engineering Research and Practice in Sweden (SERPS'03), Lund :*, 2003.

[3] MARTE specification version 1.0 (formal/2009-11-02), http://www.omgmarte.org, Accessed: August 2011.

[4] EAST-ADL Specification V2.1, http://www.atesst.org, Accessed: August 2011.

[5] OMG SysML Specifcation V1.2, http://www.sysml.org/specs.htm, Accessed: August 2011.

[6] J. Jürjens, *Secure Systems Development with UML, ISBN: 978-3-540-00701-2*. Springer, 2005.

[7] I. Weisemöller and A. Schürr, "A comparison of standard compliant ways to define domain specific languages." Berlin, Heidelberg: Springer-Verlag, 2008, pp. 47–58.

[8] B. Selic, "A systematic approach to domain-specific language design using uml," in *Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, 2007.

[9] F. Noyrit, S. Gérard, F. Terrier, and B. Selic, "Consistent modeling using multiple uml profiles," in *Model Driven Engineering Languages and Systems*, 2010.

[10] A. Borg, M. Patel, and K. Sandahl, "Good practice and improvement model of handling capacity requirements of large telecommunication systems," in *RE '06: Proceedings of the 14th IEEE International Requirements Engineering Conference*, Washington, DC, USA, 2006.

[11] CHESS Project: Composition with Guarantees for High-integrity Embedded Software Components Assembly, http://chess-project.ning.com/, Accessed: August 2011.

[12] Telecommunication Standardization Sector (ITU-T), http://www.itu.int/en/pages/default.aspx, Accessed: June 2011.

[13] M. R. Abid, D. Amyot, S. S. Somé, and G. Mussbacher, "A uml profile for goal-oriented modeling," in *Procs. of SDL'09*, 2009.

[14] T. Lodderstedt, D. A. Basin, and J. Doser, "Secureuml: A uml-based modeling language for model-driven security," in *Proceedings of the 5th International Conference on The Unified Modeling Language*, ser. UML '02, 2002.

[15] K. Alghathbar and D. Wijesekera, "authuml: a three-phased framework to analyze access control specifications in use cases," in *FMSE '03: Proceedings of the 2003 ACM workshop on Formal methods in security engineering*. New York, NY, USA: ACM, 2003, pp. 77–86.

[16] A. Cenys, A. Normantas, and L. Radvilavicius, "Designing role-based access control policies with uml," in *Journal of Engineering Science and Technology Review*, vol. 2, no. 1, 2009, pp. 48–50.

[17] K. P. Peralta, A. M. Orozco, A. F. Zorzo, and F. M. Oliveira, "Specifying security aspects in uml models," Toulouse, France, September 2008.

[18] S. Demurjian, J. Pavlich-Mariscal, and L. Michel, "Enhancing uml to model custom security aspects," in *Proceedings of the 11th International Workshop. on Aspect-Oriented Modeling*, October 2007.

[19] T. Doan and S. Demurjian, "A.: Mac and uml for secure software design," in *In: Proc. of 2nd ACM Wksp. on Formal Methods in Security Engineering*. ACM Press, 2004, pp. 75–85.

[20] S. H. Houmb and K. K. Hansen, "Towards a uml profile for security assessment," in *Workshop on Critical Systems Development with UML*, 2003.

[21] AADL, "The Architecture Analysis & Design Language: An Introduction ," http://www.aadl.info/aadl/currentsite/aadlstd.html, Accessed: August 2011.

[22] AUTOSAR Home Page, http://www.autosar.org/, Accessed: August 2011.

[23] S. Ansi, A. Albinet, S. Tucci-Pergiovanni, C. Mraidha, S. Gérard, and F. Terrier, "Completing east-adl2 with marte for enabling scheduling analysis for automotive applications," in *Embedded Real Time Software and Systems Conference (ERTS)*, Toulouse, France, 2010.

[24] TIMMO Project, http://www.timmo.org/, Accessed: August 2011.

[25] M. dos Santos Soares and J. L. M. Vrancken, "Model-driven user requirements specification using sysml," *Journal of Software*, vol. 3, pp. 57–68, 2008.

[26] S. Bernardi, J. Merseguer, and D. Petriu, "A dependability profile within marte," *Journal of Software and Systems Modeling*, 2009.

[27] E. Albrechtsen, "Security vs safety," NTNU - Norwegian University of Science and Technology http://www.iot.ntnu.no/users/albrecht/, Accessed: May 2011.

[28] A. Albinet, J.-L. Boulanger, H. Dubois, M.-A. Peraldi-Frati, Y. Sorel, and Q.-D. Van, "Model-based methodology for requirements traceability in embedded systems," in *Procs. of ECMDA'07*, Haifa, Israel, Jun. 2007.