Engineering dependability requirements for complex systems – A new information model definition

R. Guillerm and H. Demmou
CNRS ; LAAS, 7 avenue du colonel Roche, F-31077 Toulouse, France
University of Toulouse; UPS, INSA, INP, ISAE; LAAS, F-31077 Toulouse, France
{guillerm, demmou}@laas.fr

N. Sadou
SUPELEC
IETR
Avenue de la boulais, F-35511 Cesson-Sevigne
nabil.sadou@supelec.fr

Abstract—**Requirements engineering is an important phase in a system's life cycle. It is important to perform it correctly. The increasing complexity of systems makes requirements engineering activities more difficult. In design of complex system, the system engineering is widely used. Model-driven engineering, in which models are the main artifact during system development, is an emergent approach that tries to address system complexity by the intense use of models. In this context, this paper proposes a new information model based on SysML to properly manage requirements with a special attention to dependability requirements.**

Keywords- system engineering, dependability requirements, EIA-632 standard, SysML.

I. INTRODUCTION

The increased system complexity, due to the integration of different technologies such as computing, mechanic, electronics, hydraulics, etc., added to the economic constraints, make it more difficult to achieve the design of such systems. The more actual system design approach base on System Engineering (SE) concepts rely on a general methodological approach facilitating and guiding the development project [1].

If the main system needs are taken into account through SE standards, dependability needs and requirements [2] remain behind. Indeed, since dependability engineering require specific methods, and operate in parallel to 'system design'. Moreover, dependability aspects are often considered locally (component/equipment scale), or at least homogeneous component (mechanical, electronics, etc., i.e. within the same engineering).

Dependability of complex systems relies heavily on the emergent properties that result from the complex interdependencies that exist among the involved systems and their environments. It seems that the dependability properties of these systems must be addressed in an overall study, early in the design phase. This became possible with the integration of dependability analysis in the system engineering process.

One of SE processes is requirements engineering (RE) [3]. RE is generally considered in the literature as the most critical process within the development of complex systems [4], [5]. Engineering dependability requirements are of concern. A common classification proposed for requirements in the literature classified requirements as functional or non-functional [6] Functional requirements describe the services that the system should provide, including the behavior of the system in particular situations. Non-functional requirements are related to emergent system properties such as dependability attributes and response time. These Non-functional properties cannot be attributed to a single system component. Rather, they emerge as a result of integrating system components. Furthermost, non-functional requirements are also considered as quality requirements, and are fundamental to determine the success of a system.

The work presented in this paper is a part of a deployment project of system engineering. It proposes an information model that can support the system design and take into account dependability analysis. This information model is based on SysML [7] modeling and can address requirements definition and their traceability [8], [9] towards the solution elements and the V&V (Verification and Validation) elements. Dependability requirements (which are Non-functional) are integrated on RE activity of management including activities related to maintenance, such as tracing and change management of requirements.

The paper is structured into 3 main parts. The second one deals with the adopted system approach for the development. The third one describes the SysML language that we use and its interest in our approach and presents the information model.

II. SYSTEM DESIGN APPROACH

One of system design approach is the process approach. It is based on the observation that whatever the strategy used to develop a system, development activities are the same. The technical processes are based on various aspects of system engineering. They are defined by standards of System Engineering (IEEE-1220, EIA-632, ISO-15288).

The processes approach is flexible; it fits better for complex systems. Moreover, the process vision does not constrain the sequence of development activities in contrast to the development based on a particular development cycle (like V-Cycle for example). This difference is another motivation for adopting a process approach to system engineering. In this work the EIA-632 SE standard is used.

978-1-4244-5883-7/10/$26.00 ©2010 IEEE

A. System engineering

System Engineering [10] is an interdisciplinary approach, which provides concepts that make it possible to build new applications. It is a collaborative and interdisciplinary process of problems resolution, supporting knowledge, methods and techniques resulting from the sciences and experiment. System engineering is a framework which helps to define the wanted system, which satisfies identified needs and is acceptable for the environment, while seeking to balance the overall economy of the solution on all the aspects of the problem in all the phases of the development and the life of the system. SE concepts are adequate specifically for complex problems; research issues undergone can bring a solution [1].

B. EIA-632 standard

One famous standard, currently used in the industrial and military fields, is the EIA-632 [11]. This standard covers the product life cycle from the needs capture to the transfer to the user. It gives a system engineering methodology trough 13 interacting processes grouped into 5 groups, covering the management issues, the supply/acquisition, design and requirement, realization and verification/validation processes.

In parallel to the design and realization (system design processes and product realization processes), all processes are managed, evaluated and controlled by the technical management processes. Regarding the technical evaluation processes, they are available for systems analysis (eg risk analysis), requirements validation, system verifications or end products validations.

C. Requirements management

Requirements management is a crucial activity for project success [4]. Indeed, important number of documents can be produced at the system definition. Without requirements management, it seems impossible to ensure the consistency and the quality necessary for success. Statistical studies show that the success or failure of a project depends, on 40%, on the definition and the management of requirements. Requirements management allows to:

- collect requirements and facilitate their expression,
- detecting inconsistencies between them,
- validate them,
- manage requirements changes,
- link them to the rest of the project and/or the context,
- ensure their traceability.

It must also ensure that each requirement is properly declined, allocated, monitored, satisfied, verifiable, verified and justified. Figure 1 presents an overview of the requirements management of the standard EIA-632. The proposed information model is inspired from this pattern.

D. EIA-632 requirements workflow

Acquirer requirements come from a customer or user (including operators, where applicable). *Acquirer requirements* also come from a developer needing subsystems to make up an end product of a system. The latter are identified as *assigned requirements* and would have been defined by a prior application of the System Design processes. *Other stakeholder requirements*, when added to the acquirer requirements, make up a set of stakeholder requirements that are transformed into *system technical requirements*. Stakeholder and system technical requirements are identified, collected, or defined by completing the Requirements Definition Process. The *logical and physical solution representations*, *derived technical requirements*, *design solution* and *specified requirements* are defined by completing the Solution Definition Process. The derived technical requirements, logical solution representations, and system technical requirements reflect intermediate evolution states that are technical in nature, are validated, and are measurable. The design solution is verified against these requirements as reflected by the selected physical solution representation. *Specified requirements* constitute the controlled definition of the finished solution.

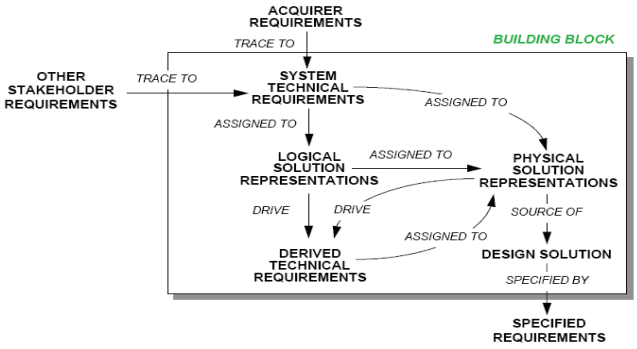

Figure 1. Requirements management of the EIA-632

E. The information model

The information model will be in the heart of knowledge of the design project, which we will be used to:

- guide the design
- manage requirements changes,
- evaluate project progress,
- or simply to help to understand the system, on the basis of a common language understandable by all.

Indeed, modeling is important for the following reasons:

- it is a support for system analysis and design,
- can be used for sharing knowledge,
- it is used to capitalize knowledge.

Modeling allows the transformation of needs into the system definition. In fact, during this transformation, we will gradually go from abstract concepts to a rigorous definition of the system. In modeling, there are 2 separate areas: the problem area and the possible solutions area. At the beginning of the project, the representation of the problem area is more important than the representation of the possible solutions area. During the progress of the design, representation of possible solutions area will be enriched to achieve the strict definition of the system. In parallel the overall representation of the problem area will be enriched to better define the expectations of the system (needs/ requirements) and will stabilize itself.

The transition between the problem domain and the solution domain is a very delicate point of system engineering. It must be expressed by allocating requirements/properties/ constraints on possible solutions. These allocations will generate traceability links which are crucial for the system verification and validation steps. We propose an information model that will be compatible with the requirements of the standard EIA-632, while adding aspects of dependability and risk management. We use SysML language to establish this information model thanks to the different available diagrams which make SysML as the language for systems engineering.

III. REQUIREMENTS MODELING AND MANAGEMENT FOR DEPENDABILITY

A. *Dependability of complex system*

Dependability is an important system-level property, and must be part of the design of these systems from the beginning. The activity of safety assessment of complex systems is inherently difficult. Dependability of these systems relies heavily on the emergent properties that result from the complex interdependencies that exist among the involved systems and their environments.

Dependability management must follow all steps of SE from the requirements definition to the verification and the validation of the system. To facilitate the dependability management from the requirements definition to system validation a SysML information model is proposed. It used for modeling and traceability analysis of requirements.

B. *SysML*

SysML is a systems modeling language that supports the specification, analysis, design, verification and validation of a broad range of complex systems [7]. The language is an evolution of UML 2.0 [12] to be applied to systems that may include hardware, software, information, processes and personnel. This may facilitate the communication between heterogeneous teams (for instance, mechanical, electrical and software engineers) that work together to develop a system. The language is effective in specifying requirements, structure, behavior, allocations of elements to models, and constraints on system properties to support engineering analysis. SysML, through a unique environment integrating requirements, allows the modeling of the design and supports different views:

- The requirements: Requirements diagram, use case diagram,
- The structure: Block diagram (internal/external),
- The behavior: Statechart, Activity diagram, Sequence diagram,
- The constraints: Parametric Diagram.

Due to the modeling coverage provided by SysML, it seems be an excellent candidate for a common language. It allows sharing specifications of a complex system between different trades, between design engineers and dependability engineers in our case.

Among other benefits of SysML, we can cite:

- Risks identification and creation of a common analytical basis to all participants of the project.
- Facilitates the management of complex projects, the scalability and the maintainability of complex systems.
- Documents and capitalizes the knowledge of all trades in a project.

SysML provides the opportunity to express the requirements using the requirements diagram. It also defines some relationships that link a given requirement to other requirements or elements of the model. It is so possible to define a hierarchy between requirements, requirement derivation, and requirement satisfaction by a model element, the requirement verification by a test case (*TestCase*) or the requirement refinement. So, this language forms a good basis for our information model. Indeed, in the system definition process, it is necessary to establish a relationship between the identified requirements and the system functions and/or components.

The traceability models linking requirements to the system components allow performing impact analysis of requirements change or modification. Thus, it is possible to assess the consequences of a requirement change on the system dependability using the network built between requirements, functions and components.

C. *Proposition*

In this part, we propose a system approach to improve requirements management for a dependable system. This approach is based on SysML information model, following the SE process of the EIA-632 standard (the processes vision is not presented in this paper). This information model is the "system" knowledge basis of the design project, allowing data sharing of data across all expertise trades (mechanical, hydraulic, thermal, electrical...). Therefore, the model is intended to model the "system" level, showing the interactions with the environment and the connections between the various subsystems.

The information model must be seen as a means of knowledge sharing, including the 3 components: requirements, design solution and V&V. It is considered as the interconnection level between all the different trades.

The safety authorities impose a separation of system design concepts [13]. The requirements, the design solution and the V&V parts must be developed independently. We must be able to distinguish clearly these different concepts.

Based on the previous observation the proposed approach allows the expression of all the concepts, while maintaining a separation between these concepts and allowing the creation of traceability link between these concepts in order to facilitate understanding and impacts analysis.

With *SysML*, it is easy and possible to mix all the concepts in a single diagram. We propose an extension of *SysML* and information meta-model that allows the structuring of the elements of the design project and the concepts separation. In other words, our approach allows a rigorous organization of the project design. Different diagrams manage different concepts.

D. *The information model*

The information model (Figure 2) that we propose is adapted to the EIA-632 standard, especially making a clear distinction between different requirements classes (acquirer, other stakeholders, technical or specified).

To achieve this meta-model in *SysML*, we have extended the language. Firstly, we define new stereotypes to requirements, while adding new attributes to our requirements. Then we define a new link type (*specify*) linking the specified requirements to model elements.

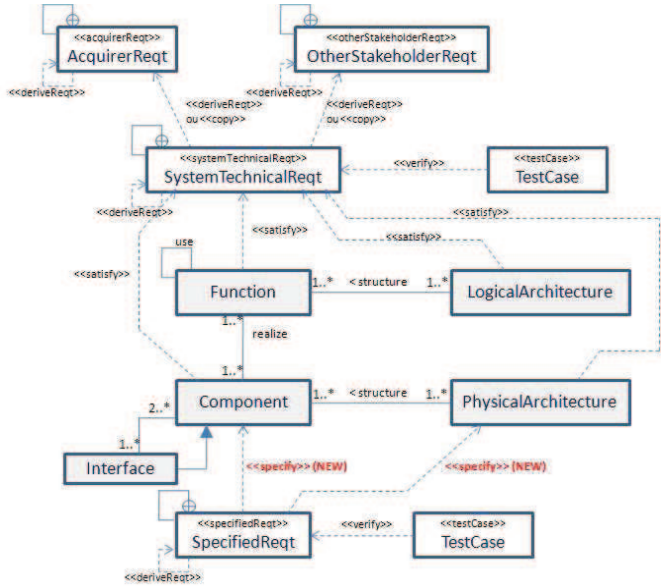

Figure 2. information model

In this model, we have simplified the number of requirements classes. Indeed, we consider that the "*systemTechnicalReqt*" represents the system technical requirements (obviously), but also the system technical requirements non-allocated to the logical solution and the derived technical requirements coming from the logical and the physical solutions.

The acquirer and other stakeholders' requirements are represented, knowing that the field 'requirement source' must be consistent with the stereotype and indicates better the concerning stakeholders in the case of "*OtherStakeholderReqt*.

All traceability links requested by the EIA-632 are taken into account in this model, and the distinction between logical solution (functional part) and physical solution (component part) appears.

In this model, we highlight the definition of interfaces, which are components themselves and which links several components together. The concept of interface is essential for a proper system design. Indeed, it is one source of problems encountered during development.

The last important element that is included in this model, neither a requirement nor a design solution element, is the "*TestCase*". These elements of V&V are included in the model to be directly connected to the requirements they satisfy.

IV. CONCLUSION

Our contributions in this paper can be summarized as follows. Firstly, we recalled the importance of dependability requirements and the criticality of the activities related to their development and management.

Dependability of complex systems is an emergent property that results from the complex interdependencies that exist among the involved systems and their environments. The second contribution is the proposition which integrates dependability analysis in the system engineering process to improve it.

To achieve properly this integration an information model, based on SysML, is proposed. This information model is the "system" knowledge basis of the design project, allowing data sharing of data across all expertise trades (mechanical, hydraulic, thermal, electrical...). Therefore, the model is intended to model the "system" level, showing the interactions with the environment and the connections between the various subsystems.

REFERENCES

[1] A.E.K Sahraoui, D. Buede, A. Sage, ``issues in systems engineering research," INCOSE congress, Toulouse, 2004

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Transactions on Dependable and Secure Computing, vol. 1, pp. 11-33, 2004.

[3] I. Sommerville, Software Engineering: (Update) (8th Edition) (International Computer Science). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[4] N. Juristo, A. M. Moreno, and A. Silva, "Is the European Industry Moving Toward Solving Requirements Engineering Problems?" IEEE Software, vol. 19, no. 6, pp. 70–77, 2002.

[5] S. Komi-Sirviö and M. Tihinen, "Great Challenges and Opportunities of Distributed Software Development – An Industrial Survey." in Proceedings of the Fifteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2003), 2003, pp. 489–496.

[6] S. Robertson and J. Robertson, Mastering the Requirements Process (2nd Edition). Addison-Wesley Professional, 2006.

[7] OMG Meta-Object Facility (MOF) Specification v. 1.4." 2002. [Online]. Available: http://www.omg.org/mda/index.htm

[8] O. C. Z. Gotel and C. W. Finkelstein, "An analysis of the requirements traceability problem," in International Conference on Requirements Engineering, 1994, pp. 94– 101.

[9] A.-E.-K. Sahraoui, "Requirements Traceability Issues: Generic Model, Methodology And Formal Basis." International Journal of Information Technology and Decision Making, vol. 4, no. 1, pp. 59–80, 2005.

[10] Systems Engineering Fundamentals. Defense Acquisition University Press, 2001

[11] EIA-632 : processes for engineering systems.

[12] G. Booch, J. Rumbaugh et I. Jacobson, The Unified Modeling Language User Guide}, Addison- Wesley, 1998.

[13] L. Boulanger, Q-D. Van. A Requirement-based Methodology for Automotive Software Development , Int. Conf. on Modeling of Complex Systems And Environments, Ho Chi Minh City, Vietnam, July 07.