System engineering approach for safety management of complex systems

R. Guillerm, H. Demmou and N. Sadou
CNRS ; LAAS, 7 avenue du colonel Roche, F-31077 Toulouse, France
University of Toulouse ; UPS, INSA, INP, ISAE ; LAAS, F-31077 Toulouse, France
rguiller@laas.fr
demmou@laas.fr
SUPELEC - IETR, Avenue de la boulais, F-35511 Cesson-Sevigne
nabil.sadou@supelec.fr

KEYWORDS
System Engineering, Complex System, EIA-632, Safety, Requirements.

ABSTRACT

This paper presents a system approach for safety management of complex system. System engineering which is an interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed is the framework of the approach. It allows taking into account the safety requirements in system engineering process to facilitates traceability of these requirements throughout the life cycle of the system. Processes of EIA-632 system standard are used to guide the proposed approach.

Introduction

The system engineering process becomes more critical as our systems increase in size and complexity. Important system-level properties, such as safety and security (1), must be built into the design of these systems from the beginning; they cannot be added on or simply measured afterward.
Systems are changed. These changes are stretching the limits of current safety engineering approaches and techniques. These changes are challenging both safety processes, methods and tools. They concern:

- Fast pace of technological change
- Changing Nature of Accidents
- New types of hazards
- Increasing complexity and coupling
- Decreasing tolerance for single accidents
- More complex relationships between humans and automation
- Changing regulatory and public views of safety

Rasmussen has argued that major accidents are often caused not by a coincidence of independent failures but instead reflect a systematic migration of organizational behavior to the boundaries of safe behavior under pressure toward cost-effectiveness in an aggressive, competitive environment (2).
Weaknesses of the current safety processes (figure 1) can be resumed in the following points (non exhaustive list):

- Safety analysis involve some degree of intrinsic uncertainty. So, there is a degree of subjectivity in the identification od safety issues.
- Different groups need to work with different views of the system (e.g. systems engineers view, safety engineers view). This is generally a benefit but it can be a weakness if the views are not consistent.
- Definition of the safety requirements and their formalization.
- Traceability of safety requirements.
- Existing / traditional safety analysis techniques are difficult to use on modern, complex systems.
- Textual description of failure modes is often too ambiguous.
- System models are developed in electronic form, but no use is made of this for Safety/ Reliability analysis. Ideally there should be a common repository of all requirements, design and safety information.

Some points are due to the absence of a safety global approach. Indeed, safety must be addressed as global property and safety requirements (3) must be formulated not only in the small but in the large.
For example, the Ariane 5 and Mars Polar Lander losses are examples of system accidents. In both of these accidents, the components did not fail in terms of not satisfying their specified requirements. The individual components operated exactly the way the designers had planned the problems arose in the unplanned or misunderstood effects of these component behaviors on the

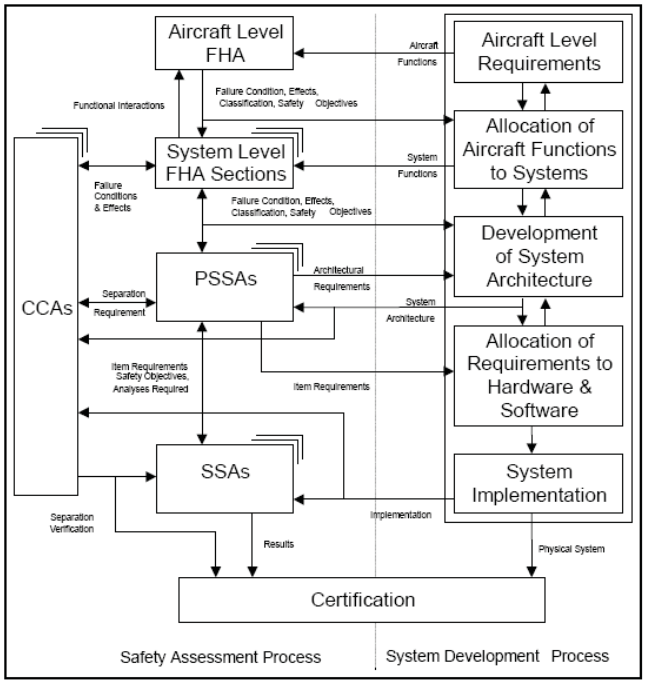

Figure 1: Safety integration

system as a whole, that is, errors in the system design rather than the component design, including errors in allocating and tracing the system functions to the individual components. The solution, therefore, lies in systems engineering. A global approach is so necessary. Indeed, safety is clearly an emergent property of systems.

Some of these points are addressed by ESACS and ISAAC projects. ESACS project (4) developed a methodology and a platform that helps safety engineers automating certain phases of their work.

The ESACS platform can be used as a tool to assist the safety analysis process from the early phases of system design to the formal verification and safety assessment phases. It gives a partial response for the weakness cited above but it focused on the use of formal methods for safety assessment and does not propose a global approach do achieve it. For example, traceability of safety requirements and Human risk analysis are not considered.

ISSAC European project (5) which is the continuation of ESACS project proposes to take into account human errors analysis. It is achieved by injecting human errors in the formal model.

Nevertheless, these two projects are essentially concentred on formal method for safety assessment and not really in a global approach to achieve it.

ASSERT is another project, but, like above projects (ESACS and ISSAC), it focused on the method and tool. Moreover, only software failures are considered.

The norme IEC 1508 (6), (7) consider the overall lifecycle. It is considered as for the management of the safety throughout the entire life of the system, but it concerns only systems that require safety functions. It is guide for the implementation of the relevant safety functions. This work is part of a project in deploying System Engineering (SE)(11) (12) . We address the integration of safety management in system engineering process. The paper is structured into five remaining parts. The second part gives a brief introduction of the emerging discipline of system engineering in matter of key processes and the standard EIA-632 (13). The third part presents briefly the integration approach. In the forth part, an original approach for safety integration in system engineering process is proposed.

The system engineering framework for complex system development

System Engineering 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 (11).

System engineering concepts

System engineering is the application of scientific and engineering efforts in order to:

- Transform an operational need into a description of system performance parameters and a system configuration through an iterative process of definition, synthesis, analysis, design, test and evaluation.
- Integrate reliability, maintainability, availability, safety, survivability, human engineering and other factors into the overall engineering effort to meet cost, schedule, supportability and technical performance objectives.

System engineering is an interdisciplinary approach that:

1. Encompasses the scientific and engineering efforts related to development, manufacturing, verification, deployment, operations, support and disposal of systems products and processes.

2. Develops needed user trainings, equipments, procedures and data.

3. Establishes and maintains configuration management of the system.

4. Develops work breakdown structures and statements of work and provides information for management decision-making.

System engineering is a management methodology to assist designer through the formulation, analysis and interpretation of the impacts of proposed policies, controls or complete systems upon the need perspectives, institutional perspectives and value perspectives of stakeholders to issues under consideration.

System engineering is an appropriate combination of the methods and tools of a suitable methodological process and systems management procedures.

We distinguish three levels in System engineering as illustrated in figure 2. The first level, SE processes, focus on high-level issues, high-level requirements such as business needs and strategic needs and methods.

The second level, System engineering methodologies and methods, deals with all technical issues such as systems requirements design methodologies standards.

The third level, System engineering tools or technologies, covers the implementation issues concerning the tools to be used, the required technologies to respond to the various assets of requirements such as reliability, costs, maintainability and enabling technologies.

System engineering assists designer who desire to develop policies for management, direction, control and regulation activities relevant to forecasting, planning, development, production and operation of total systems Figure 2.

In System engineering best practice, we have the following chain:

Processes $\rightarrow$ Methods $\rightarrow$ Tools

These entities, such as processes, methods and tools, are the conceptual basis of our approach taken from System engineering best practice. In the first step, the processes can be identified with respect to the accumulated know-how, and can also be taken from a standard as the thirteen generic processes proposed in standard EIA-632. The second step concerns the methods to be used. The methods can be either developed or used by the existing one, which implement the process as we cannot choose a method for its flexibility or popularity but only if it reflects the semantics of the process. No taxonomy has yet been developed for corresponding processes and methods. The third step concerns the tools that do not correspond to the processes but the methods; hence in this approach we cannot use a tool to implement a process without first identifying the associated methods.

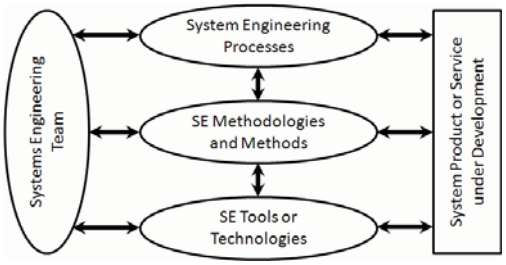

Figure 2: Three levels of system engineering.

EIA-632 standard

One famous standard, currently used in the industrial and military fields, is the EIA-632. 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. Figure 3 shows the interaction between all the 5 groups of processes, whose roles are (13):

1. Technical management processes (three processes): these processes monitor the whole process ranging from the initial idea of building a system until its delivery.

2. Acquisition and supply processes (two processes): these processes ensure the supply and acquisition (and are very close to logistics).

3. System design processes (two processes): these processes are on the elicitation and acquisition of requirements and their modelling, the definition of the solution and its design.

4. Product realization processes (two processes): these processes deal with the implementation issues of system design and its use.

5. Technical evaluation processes (four processes): these processes deal with verification, validation and testing issues.

Briefly, the operation of the proposed processes is:

- One acquisition request arrives and is treated by the supply process by establishing an agreement,

- The acquirer requirements are then transmitted to the System Design processes in charge of the elaboration of the logical solution, then the physical one, and also lots of sets of specified technical requirements, where each set is associated to a sub-system.

- The acquisition process is in charge to buy (if available in the market) or to make build the sub-systems responding to the different sets of specified requirements.
- Once the sub-systems received, the realization of the final product can begin, based on the design solution previously established and chosen.
- To finish, the final system will be transferred to the user, just after tests and final validation.

In parallel, all the previous processes are managed, rated and controlled by the technical management processes. And the technical evaluation processes allows to do system analysis (like risk analysis), requirement validations or system verification, during the development and when needed.
In fact, one or several sub-processes are defined for each 13 processes and the developer should decide which of the all 33 sub-processes apply. In this paper, only the system design processes and the technical evaluation processes are considered. These processes appear as the most important for safety management.

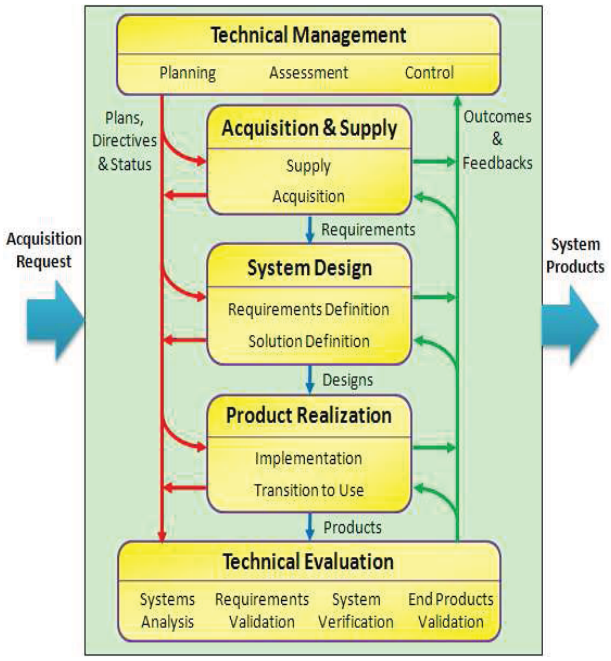

Figure 3: System engineering processes

Safety in system engineering process

Safety

Safety is an important system-level property, and must be built into the design of these systems from the beginning.

The safety assessment process can be decomposed into three main phases:

- preparation phase which initiates the assessment (Safety Target are defined)
- conduct phase in which the assessment is performed
- conclusion phase in which the assessment results are delivered.

System Engineering is the ideal framework for the design of complex system. In this work It is considered as a framework to manage safety.
A system engineering approach to safety starts with the basic assumption that the safety propriety, can only be treated adequately in their entirety, taking into account all variables and relating the social to the technical aspects (9). This basis for system engineering has been stated as the principle that a system is more than the sum of its parts.
The Safety management must follow all steps of SE From the requirements definition to the verification and the validation of the system. If we consider, for example, a reliability requirement defined for a global system, its formalization and analysis must allow ensuring that the technical solutions selected with design progression deals with this reliability requirement at sub-systems level and after their integration.
Note that this paper illustrates the proposed approach in term of process which must be defined independently to methods and/or tools (other projects which are focused on the methods and tools (4) and (5) for example). These different works will be exploited in the safety management from a global point of view.

Integration approach

The integration of safety must concern all system engineering processes. This paper is focused only on:

- System Design processes,
- Technical Evaluation processes.

The safety requirements must be taken into account in requirements definition process. It allows the formulation, the definition, the formalization and the analysis of these requirements. Then a traceability (10) model must be build to ensure the taking into account of the requirements throughout the development cycle of the system.
These Safety requirements influence acquirer requirements, stakeholder requirements, system technical requirements, logical solution representations and physical solution representations.
Technical Evaluation processes define 12 types of sub-processes going from requirement statements validation to enabled product readiness. The sub-processes (The

task associated to each sub-process can be consulted in (13)) considered are:

- requirements statements validation,
- acquirer requirements validation,
- other stakeholder requirements validation,
- system technical requirements validation,
- logical solution representations validation,
- design solution verification.

The implementation of the approach consists in identifying and indicating in which way the safety must be considered for each sub-processes of EIA-632. In other words, the sub-processes of EIA-632 standard are translated or refined in terms of safety and included in system design process.

EIA-632 sub-process to safety refinement

In this section we address EIA-632 processes with safety point of view.

System design processes

The System Design Processes are used to convert agreed-upon requirements of the acquirer into a set of realizable products that satisfy acquirer and other stakeholder requirements.
Two processes are involved: the Requirements Definition Process and the Solution Definition Process. The relationship between these two processes is shown in Figure 4.

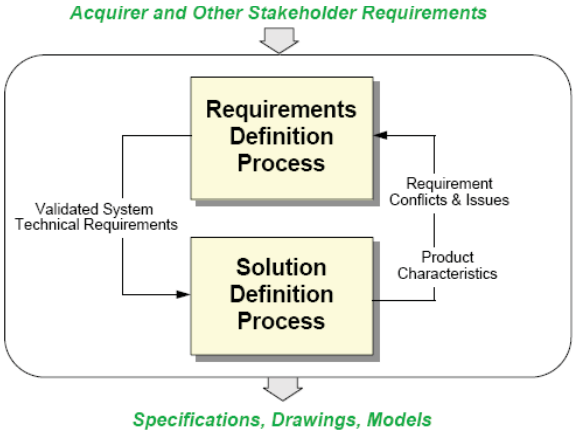

Figure 4: System design processes

Requirement definition process
The goal of the requirements definition process is to transform the stakeholder (the acquirer and all other stakeholders who have an interest in the system) requirements into a set of technical requirements. For functional and no-functional requirements, if this distinction is not possible at the requirement elicitation process level, the analyzer may do it to categorize requirements. To perform this task, 3 sub-processes are associated with this process: the Acquirer Requirements, the Other Stakeholder Requirements and the System Technical Requirements sub-process. The Requirements Definition Process is re-accomplished, if necessary.

a. Acquirer Requirements
The developer shall define a validated set of acquirer requirements for the system, or portion thereof.
In the safety framework, acquirer requirements, generally, correspond to constraints in the system. It is necessary to identify and collect all constraints imposed by acquirer to obtain a dependable system. A hierarchical organization associates weight to safety requirements, following their criticality.
Some Standards are available to guide designer to define safety requirements. For example, for safety critical systems within the civil aerospace sector are developed subject to the recommendations outlined in ARP4754 (14) and ARP4761 (15). These standards give guidance on the 'determination' of requirements, including requirements capture, requirements types and derived requirements.
When the requirements are defined (16)some attributes can be used to facilitate their management.

b. Other Stakeholder Requirements
The developer shall define a validated set of other stakeholder requirements for the system, or portion thereof.
The same approach applied to acquirer requirements is applied to Other Stakeholder Requirements.

c. System Technical Requirements
The developer shall define a validated set of system technical requirements from the validated sets of acquirer requirements and other stakeholder requirements.
System technical requirements must be unambiguous, complete, consistent, achievable, verifiable, and necessary and sufficient for a system design.
For safety requirements, the system technical requirements traduce system performances. It consists on defining safety attributes (5. Determine risk tolerability, MTBF, MTBR, failure rate for example).
Among all requirements formulated by the acquirer and other stakeholder, some of them are safety requirements. For example, critical events define reliability requirements in the sense that their taking into account must leads to a design of a system able to avoid these events.

Solution Definition Process
The Solution Definition Process is used to generate an acceptable design solution. This solution satisfies:

1. the system technical requirements resulting from the Requirements Definition Process,

2. the derived technical requirements from the Solution Definition Process.

Three sub-processes are associated with the Solution Definition Process.

a. Logical Solution Representations
The developer shall define one or more validated sets of logical solution representations that conform with the technical requirements of the system. In order to do this, he must in particular (1) do some tradeoff analysis, (2) identify and define interfaces, states and modes, timelines, and data and control flows, (3) analyze behaviors, and (4) analyze failure modes and define failure effects.
Formal models can be used for logical solution representations. The use of formal methods allows for automation of verification and analysis and for a tighter integration between system design and safety analysis. The model can automatically enriched with failures in order to perform safety analysis.

b. Physical Solution Representations
The developer shall define a preferred set of physical solution representations that agrees with the assigned logical solution representations, derived technical requirements, and system technical requirements.
The physical solution representations are derived from logical solution representation and must respects all requirements, particulary, safety requirements.

c. Specified Requirements
These requirements concern the design solution. The designer must ensure that the design solution is consistent with its source requirements. The safety analysis process allows the validation of these requirements.

Technical Evaluation Processes

The Technical Evaluation Processes are intended to be invoked by one of the other processes for engineering a system. Four processes are involved: Systems Analysis, Requirements Validation, System Verification and End Products Validation. The relationship between these processes is shown in Figure 5.
In this paper, we focus only on 3 processes of the technical evaluation:

1. Systems Analysis Process, which contains a Risk Analysis sub-process,

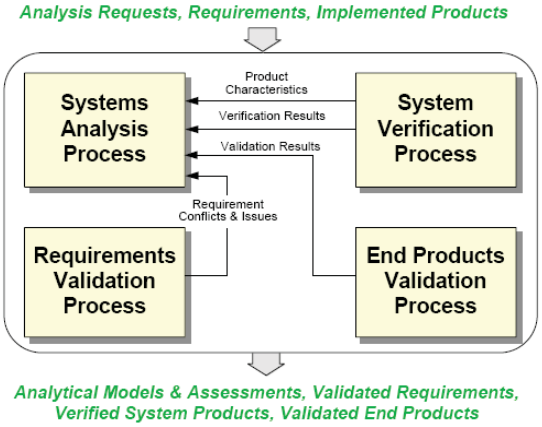

Figure 5: Technical evaluation processes.

2. Requirements Validation Process,

3. System Verification Process.

Systems Analysis Process
The Systems Analysis Process is used to:

1. Provide a rigorous basis for technical decision making, resolution of requirement conflicts, and assessment of alternative physical solutions;

2. Determine progress in satisfying system technical and derived technical requirements;

3. Support risk management;

4. Ensure that decisions are made only after evaluating cost, schedule, performance, and risk effects on the engineering or reengineering of the system.

a. Risk Analysis sub-process
The developer shall perform risk analysis to develop risk management strategies, support management of risks and support decision making.
Several techniques can be used to analyze risks, for example: fault tree, or Failure Mode, Effect, and Criticality Analysis. This step is very important, because it determines the risks of the system.
The step of risk analysis can generate safety requirements other than that defined by the acquirer and stakeholder. These new requirements must be taken into account.

Requirements Validation Process
Requirements Validation is critical to successful system product development and implementation. Requirements are validated when it is certain that they describe the input requirements and objectives such that the resulting system products can satisfy them. The Requirements Validation Process helps to ensure that

the requirements are necessary and sufficient for creating design solutions appropriate to meet the exit criteria of the applicable engineering life cycle phase and of the enterprise-based life cycle phase in which the engineering or reengineering efforts occur. In this process, a great attention is done to traceability analysis, which allows verifying all the links among Acquirer and Other Stakeholder Requirements, Technical and Derived Technical Requirements, and Logical Solution Representations.
Like other requirements, safety requirements must be validated. The validation allows to design safe system. to facilitate this step, semi-formal solutions, like UML (17) or SysML (18) (which is an UML profile for systems engineering), can be used for good formulation of requirements. Indeed the diversity of people concerned by the system design project can have limited knowledge concerning the structure of a future system makes industry-scale requirement engineering projects so hard. So the UML or SysML with their different diagrams can be helpful.

System Verification Process
The System Verification Process is used to ascertain that:

1. The generated system design solution is consistent with its source requirements.
2. End products meet their specified requirements at each level of the system structure implementation (from the bottom up).
3. Enabling product development or procurement for each associated process is properly progressing.
4. Required enabling products will be ready and available when needed to perform.

Simulation is a good and current method used to achieve system verification. Other methods like virtual prototyping, model checking and other ones can be used.

Conclusion

The approach presented in this paper concerns safety integration in system engineering process. It allows to give some guidelines to address efficiently safety of complex systems in all phase of system design. The approach is based on EIA-632 standard and can be resumed as follows:

- Elements of the approach: process → methods → tools
- System engineering processes handled using EIA-632 standard
- Integration of safety analysis in system engineering process

The paper addresses some processes of the standard EIA-632 independently to methods and/or tools. The processes addressed are system design process and technical evaluation process.
The work is in progress and other processes will be used and defined in the integration approach. The next step of the deploying system engineering project is to propose appropriate methods and tools for achieving each subprocesses of the EIA-632 standard.

REFERENCES

[1] 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.

[2] Jens Rasmussen. Risk Management in a Dynamic Society: A Modelling Problem. Safety Science, vol. 27, No. 2/3, Elsevier Science Ltd., 1997, pp. 183213.

[3] Finkelstein, A. (1993). Requirements Engineering: an overview. 2nd Asia-Pacific Software Engineering Conference (APSEC'93), Tokyo, Japan, 1993

[4] M. Bozzano, A. Cavallo, M. Cifaldi, L. Valacca, A. Villafiorita. FM 2003. Pisa, 8-14 September 2003

[5] O. Akerlund, P. Bieber, E. Boede, M. Bozzano, M. Bretschneider, C. Castel, A. Cavallo, M. Cifaldi, J. Gauthier, A. Griffault, O. Lisagor, A. Ldtke, S. Metge, C. Papadopoulos, T. Peikenkamp, L. Sagaspe, C. Seguin, H. Trivedi, L. Valacca. ISAAC, a framework for integrated safety analysis of functional, geometrical and human aspects. European Congress on Embedded Real-Time Software (ERTS 2006), Toulouse, 25, 26, 27/01/06

[6] Felix Redmill, Redmill Consultancy. An Introduction to the Safety Standard IEC 61508. Journal of the System Safety Society, Volume 35, No. 1, First Quarter 1999

[7] J. Brazendale. IEC 1508: Functional Safety: Safety-Related Systems. Software Engineering Standards Symposium, 1995.

[8] Richard C. Booten Jr. and Simon Ramo. The development of systems engineering. IEEE Transactions on Aerospace and Electronic Systems, AES-20(4):306309, July 1984.

[9] K. Kotovsky, J.R. Hayes, and H.A. Simon. Why are some problems hard? Evidence from Tower of Hanoi. Cognitive Psychology, vol. 17, 1985.

[10] Gotel, O. and Finkelstein, A. (1994). An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering (ICRE'94), Colorado Springs, April 1994, pp. 94-101.

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

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

[13] EIA-632 : *processes for engineering systems.*

[14] Certification considerations for highly-integrated or complex aircraft systems. Society of Automotive Engineers, December 1994.

[15] Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. Society of Automotive Engineers, August 1995.

[16] JL. 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.

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

[18] SysML: Source Specification Project, http://www.sysml.org/