A Goal-Oriented Approach for Safety Requirements Specification

Elena Navarro[†], Pedro Sánchez[‡], Patricio Letelier[∫], Juan A. Pastor[‡] and Isidro Ramos[∫]
[†]*Computer Science Department, UCLM*
[‡]*Information Technology and Communication, UPCT*
[∫]*Department of Information Systems and Computation, UPV*
[†]*enavarro@info-ab.uclm.es*
[‡]*{pedro.sanchez| juanangel.pastor }@upct.es*
[∫]*{letelier|iramos@dsic.upv.es*

Abstract

Robotic systems are developed to execute tasks with several types of risks associated. The possible damages that can affect both the working environment and the self-system lead us to consider that these systems are safety critical, i.e., systems where the strict management of safety aspects is vital. In this work we introduce our proposal for the consideration of safety related requirements and their consequent trace to the desired final system architecture. For this reason, this paper gives a procedure for the identification and specification of safety requirements based on a goal oriented framework. Moreover, in this work other approaches have been considered and integrated to deal with well known safety standard recommendations. By means of an industrial case study, we show how this proposal can be used to consider safety requirements in tele-operated robotic systems and, by extrapolation, in other critical domains.

1. Motivation

Robotics systems are substantially different from other software applications because it is mandatory to consider aspects such as interaction with the environment, presence of perturbations, etc. Moreover, if their use has an important impact on persons and equipment when errors arise then safety aspects must be considered during their development. Besides the risks inherent to their use, work places are higher risk areas. Douglass [2] gives, among others, the following list of damage sources: errors in the execution of the control system (hardware and software), people who access to forbidden walking areas, human errors, broken mechanical parts, liberation of stored energy, and so on.

Nowadays, when we analyze the development of tele-operated systems, we can observe that there is no well known integration of safety requirements within the process of requirements specification as whole, nor are there methodologies supporting it in an integrated way. To meet this deficiency, we have considered the *goal oriented* framework ATRIUM [16] for guiding the requirement engineering process. ATRIUM offers a methodology for defining both the requirement specification and the software architecture of the system. This framework has been extended for considering *safety* requirements by following the ANSI/RIA R15.06-1999 [1] standard. Furthermore, our proposal has been enriched both with the ideas by Lemos [11] for the division of operation mode of systems, and the patterns and heuristics given by Douglass [2] for the consideration of this kind of requirement.

This work has been validated in the context of the EFTCoR project [4] funded by the European Union. This kind of system, by its nature, entails a greater probability of dangers than others. Thus a precise identification, specification and trace of safety requirements turn out essential.

The remainder of the paper is organized as follows: In Section 2, we present the conceptual framework considered. In Section 3, we describe the process of specifying a complete requirements specification, integrating safety requirements, by following a goal oriented approach. The case study is introduced in Section 4 by explaining the real situation where these ideas have been put into practice. Section 5 gives an example of pattern use for the materialization of safety requirements. Section 6 looks at some related work and finally the paper is concluded in Section 7 with future work.

Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06)
0-7695-2546-6/06 $20.00 © 2006 IEEE

2. Conceptual Framework

In order to clarify the following discussion, the main concepts used are defined in this section. Hence, it is mandatory to define which meaning of *Safety requirement* has been used from the set of definitions in the bibliography. ANSI/RIA R15.06-1999 standard, which this work is based on, defines Safety requirements as those to be satisfied for any industrial robotic system to assure the safety of personnel associated with its use. In this work, this definition has been extended, with Levesons' ideas [15], by including damage or destruction of property or injury or damage to any living being, especially, human beings.

When the behaviour of a system is being described, *tasks* to be provided by any system component, whether software or hardware, have also to be described. They mainly refer to that functionality expected from the system, as for instance, motions of any element of the robot, coordinated motions, use of tools, etc. The execution of these tasks is the potential source of damage or injuries the system can cause, so that the most serious risks for safety arise from deficiencies of functionality, reliability or usability as ISO/IEC 9126 standard [9] states.

The early detection of *Hazards* is a challenge during the specification of Safety requirements. A hazard is any potential source of damage or injury to an entity of the system, from an operator to an entity of the environment or the self system. In this way, during the gathering and specification of Safety requirements, the analyst has to identify the likelihood of the system hazards and analyze them so as to determine which strategy is the most appropriate for their management. For this analysis, the risks related to each hazard have to be established, i.e., the damage or injury to any entity.

These concepts indicate that although the process of gathering and identifying this kind of requirement is quite similar to that applied in other contexts, it does show some meaningful differences. Furthermore, the fact that tele-operated systems are Safety Critical emphasizes how important it is to provide the analyst with a both specific process and specific notation.

A *Goal* describes why a system is being developed, or has been developed, from the point of view of the business, organization or the system itself. In order to identify it, both functional goals (expected services of the system) and non functional goals (quality of service, constraints on the design, etc.) should be determined. A *Goal Model* is built as a directed graph by means of a refinement from the systems goals (or concerns). This refinement lasts until goals have enough granularity and detail so as to be assigned to an agent (software or environment) so that they are verifiable within the system-to-be. This refinement process is performed by using AND/OR/XOR refinement relationships. In addition, operationalizations are also specified during the Goal Model definition. Operationalizations are the lowest level refinements introduced to describe the design alternatives associated to the requirements by means of contribution relationships.

This approach offers two advantages that make it especially appropriate for analyzing the specification. The first advantage is related to its ability to specify and manage positive and negative interactions among goals, which allow the analyst to reason about different design alternatives and to validate the Goal Model by means of its animation. The second advantage is related to its capabilities concerning traceability from low-level details to high-level goals (concerns), which make it especially suitable to bridge the gap between architectural and requirements models.

3. Identifying and Specifying Safety Requirements

Both ANSI/RIA R15.06-1999 standard and Douglass' and Lemos' approaches have been integrated to establish a process for identification and specification of Safety Requirements in Tele-operated environments. The Goal Model of ATRIUM is the notation used during the process. This model exploits the standard ISO/IEC 9126 as a starting point to organize the requirements specification.

The established process entails several steps. The first ones (I-II) are related to the behaviour specification of the system:

I. *To identify system operation modes*, according to Lemos' recommendations for control systems. These operation modes are specified as system goals and have a refinement relationship towards the characteristic ISO/IEC 9126 Suitability. Each goal is related to its child sub-goals by means of AND/OR/XOR refinement relationships. The relationship to be applied depends on whether all sub-goals, some of them or only one sub-goal, respectively, have to be included in the system to satisfy the parent goal. This intentional refinement is applied over and over again until goals have enough granularities to allow the identification of tasks.

II. *To identify Tasks* (T_i) *associated to each operation mode*. In the Goal Model, these tasks are specified as requirements which have an AND/OR/XOR refinement relationship towards their goal parent. In addition, it provides an improved visibility of which hardware/software components of the system are involved in which operation modes because of their traceability to operationalizations, i.e., scenarios which describe how these components interact to fulfil a requirement.

Both the identification of operation modes and tasks provide the system behaviour specification. This identification must be performed before Safety requirements are identified, so that any likely injury or damage is caused when the system is performing these tasks. Once this view of the system has been established, Safety requirements are determined by using the ISO/IEC 9126 Safety category. Therefore, the following steps are added to the process:

III. *Determine Safety Goals of the system*. For each identified task T_i its capacity to cause any damage is evaluated. If this happens, a Safety goal, *Safe* (T_i), will be specified so as *to achieve safeguard* T_i. This implicitly means a composition relationship between a Safety requirement *Safe* (T_i) and a Functional requirement T_i at the task level.

IV. *Determine System Hazards*. For each safety goal *Safe* (T_i) its related hazards are identified and specified as a sub-goal *Manage* (Hz_j). An AND/OR/XOR relationship is established between *Safe* (T_i) and its set of hazards to be managed. The relation to apply depends on whether the whole set of hazards, some of them or only one, respectively, have to be managed to safeguard the task T_i. We have to bear in mind that the same hazard can be specified as refinement of several Safety goals.

V. *Identify Risks* (R_k) *associated to each pair Safe* (T_i)—*Manage* (Hz_j). The associated risk R_k has to be identified, so that the strategy appropriate for the management of Hz_j is selected. A Hazard can be related to several risks depending on which task is to be evaluated. Hence, according to the framework of traceability [13], the risk specification is represented by introducing a *rationale* associated to the refinement relationships between *Safe* (T_i)—*Manage* (Hz_j). In addition, according to Douglass' evaluation of risks, this rationale also entails some necessary information for the risk specification: identification of causes (software, hardware, human, …) which can result in a hazardous situation, reaction required for its management and maximum deadlines of exposure, detection and tolerance.

VI. *Determine the Risk Reduction Category (RRC) to apply to each relation Safe* (T_i)—*Manage* (Hz_j), taking into account its associated risks R_1 ... R_n. With this aim, the following three attributes must be evaluated:

a) *Severity*. Level of damage that an entity of the environment or the self system can suffer. The table which is provided by ANSI/RIA R15.06-1999 for evaluation has been modified so as to deal with the widest sense of the term, i.e., including not only damage to the health or the environment but also to the self system. Therefore, the meanings associated to both S1 and S2 categories are described now as follow:

S2	Serious injury to the operator requiring more than first-aid. *Damage of a system component which is irreplaceable both in time and cost.*
S1	Serious injury to the operator only requiring first-aid. *Damage of a system component which is replaceable both in time and cost.*

b) *Exposure.* Frequency of exposure to the hazard. ANSI/RIA R15.06-1999 defines two categories: *E2* as *frequent* and *E1* as *infrequent*.

c) *Avoidance*. Likelihood of avoiding the exposure to the hazard. ANSI/RIA R15.06-1999 defines two categories: *A2* as *not likely* and *A1* as *likely*.

ANSI/RIA R15.06-1999 describes eight combinations of these values, which are specified in Table 1 as Risk Reduction Categories (RRCs) along with the recommended actions to manage the Hazard. Hence, a evaluation of the values (Severity, Avoidance, Exposure) is carried out on each pair Safe (Ti)—Manage (Hzj) and its risks Rk. According to the RRC an action has to be selected to eliminate, substitute, prevent, isolate or cease the Hzj. If different RRCs are applicable for the same pair Safe (Ti)—Manage (Hzj), the most severe will be the selected one.

Once each pair *Safe* (T_i)—*Manage* (Hz_j) has been dealt with along with its Strategy, it is mandatory to re-evaluate the full set of RRC for the Goal Model as the set of Hazards could have changed. This procedure must be repeated until all the hazards are considered "tolerable", i.e., an acceptable risk level for the system. These hazards are to be considered residual risks of the system.

Table 1 Risk Reduction Categories

Exposure	Avoidance	Severity	RRC	Action on Hz
E2	A2	S2	R1	Eliminate/Substitute
		S1	R2C	Prevent/ Cease
	A1	S2	R2A	
		S1	R3A	Isolate
E1	A2	S2	R2B	Prevent/Cease
		S1	R3A	Isolate
	A1	S2	R3B	
		S1	R4	Warning/Training/Protect

4. Case Study

This section illustrates how we have applied our proposal to gather the safety requirements of the industrial project EFTCoR. The aim of this project is to design a family of robots for performing ship hull maintenance operations such as coating removal, washing and re-painting.

The identified robotic tele-operation platform is integrated by different subsystems (illustrated in Figure 1). The Robotic Devices Control Unit (RDCU) integrates all the required functionality to manage the EFTCoR. Cleaning Tools and Positioning Systems, both Primary and Secondary, are the mechanical components of the EFTCoR. Its architectural definition is highly relevant because of the constraints that have to be satisfied in order to allow a safe behavior of the system.

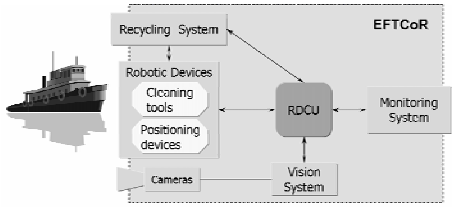

Figure 1. EFTCoR System

During the elaboration of the specification, 6 operation modes were detected (step I): *working*, *calibration*, *learning*, *diagnosis* and *configuration* (these four are *maintenance* modes) and *safe stop*. For each operation mode the involved subsystems and associated task were defined (step II). Due to limited space, in this work we only include the Working Mode and the Primary Positioning sub-system of the EFTCoR system.

Once the operation modes of the system were established, their safety requirements were detected. With this aim, those tasks to be safeguarded were specified as Safety goals, *Safe* (T_i), of the system (step III). Figure 2 (whose symbols are described in Table 5), shows a part of the Goal Model where tasks (Table 2) to be safeguarded are identified. The Goal Model for the EFTCoR, partially shown in Figure 2, has been made by means of MetaEdit+[10], a metaCase for domain-specific modelling. Each Safety Goal, *Safe* (T_i), was refined in *Manage* (Hz_j) sub-goals by applying step IV. A summary of Hazards for the Primary Subsystem is shown in Table 3.

Table 2. Summary of Tasks to safeguard

Task	Description
T1	Motion of the arm joint
T2	Motion of the joint on tracks
T3	Stop of the primary motion

Table 3. Summary of Hazards for the Primary Subsystem

Task	Description
H1	Primary Positioning subsystem is moving and finds an obstacle on the rail.
H2	End of range of the Primary joint is overrun (arm).
H3	End of range of the Primary joint is overrun (on rails).
H4	Joint of Secondary touches the hull of ship.
H5	Primary (arms or tracks) does not stop.
H6	Robot places the tool point out of the cleaning stereo-radio when the robot is in working mode.

Applying step V, the risks (described in Table 4 and as comments in Figure 2) of each pair *Safe* (T_i) — *Manage* (Hz_j) were established. Along with their identification, a description of other information is also required for the later evaluation activity. For instance, R8 and R10 were established as likely risks for the pair Safe (H6)-Manage (T2) and, *Sofware Error of the robot control* as a possible cause of hazards; the realization of an *Emergency Stop* as a possible reaction when hazard H6 appears; probability of occurrence *medium*; etc.

Table 4. Summary of Risks for the Primary Subsystem

Risk	Description
R1	Damage of the tool or any mechanical component of the Secondary Positioning subsystem.
R2	Mechanical damage to the arm joint or joint on tracks.
R3	Damage to the hull surface.
R4	Damage to operator, primary or any object of the environment.
R5	Mechanical damage of the primary with joint on tracks and overturned of the robot.
R6	Mechanical damage to Primary Joint either on tracks or arm.
R7	Damages to Secondary.
R8	Damages to operator.
R9	Mechanical damages.
R10	Damage to objects of the environment.

Table 5. Explanation for symbols used in Figure 2

Symbol	Goals on Figure 1
Ti	Safe (Ti)
⚠Hzj	Manage (Hzj)

As a result of step VI, several reduction categories were selected for use in each specific case. For instance, the evaluation of severity, exposure and avoidance was (S2, E2, A2) for the pair Safe (H6)-Manage (T2), taking into account the risks (R8, R10). By applying the RRCs shown in Table 1, the category R1 was established and *eliminate* was the selected action. Once the hazard H6 is eliminated the re-evaluation of the graph it is necessary in order to determine if the residual risks are tolerable for the system.

5. Safety Patterns For Operationalizations

Table 6 describes the strategy for the entry R2A of the safeguard selection matrix (Table 1). This recommendation is not enough to develop concrete systems such as EFTCoR and in consequence more detail is needed in order to facilitate this process. Thus, we propose to establish a catalogue of patterns for the reduction or elimination of risks, classified by the ANSI RRC. These patterns, in a similar way to the proposal by Gamma [6], describe the solution to recurrent situations. In order to facilitate its comprehension, we give a concrete example of a pattern considered in the context of the EFTCoR system.

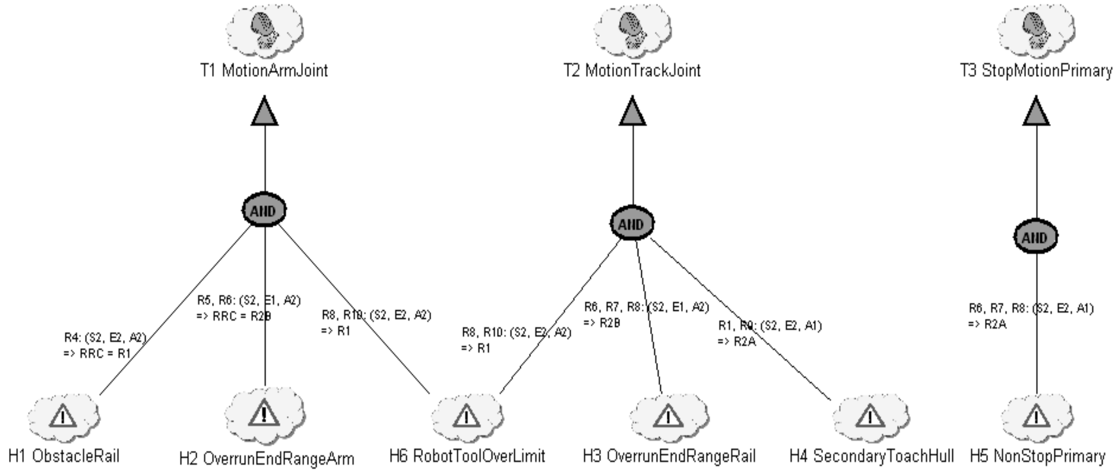

Figure 2. Goal Model (Part of)

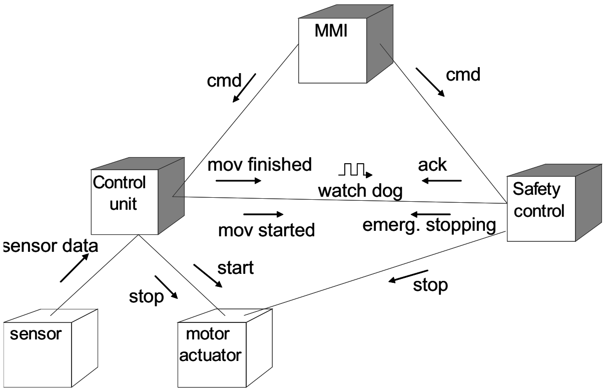

Figure 4. Deployment considering safety for hazard H5

Table 6. Description of the ANSI Risk Reduction Category R2A

R2A: Control reliable safety circuitry (based on hardware or software controller or firmware) – The monitoring shall generate a stop signal if a fault is detected. A warning shall be provided if a hazard remains after cessation of motion – Safe state shall be maintained until the fault is cleared. – Common mode failures shall be taken into account. – The single fault should be detected at time of failure.

The primary positioning system (see Figure 3) has a height of twelve meters and a weight of twenty tons which make inevitable the movement of the robot without the consideration of safety requirements. The crane has in its central zone an articulated arm of two tons with a secondary positioning system at its end (an XYZ-table which includes a cleaning tool). It is indispensable that the system ensures a safe movement of the arm according to the received commands from the operator. A detailed analysis of the hazard H5 ("the arm of the primary system does not stop") leads us to associate the following sources of error:

- Any sensor integrated with the motors that move the arm fails.
- The electrical power is off.
- The control unit does not run correctly (a hardware fail or a software error).

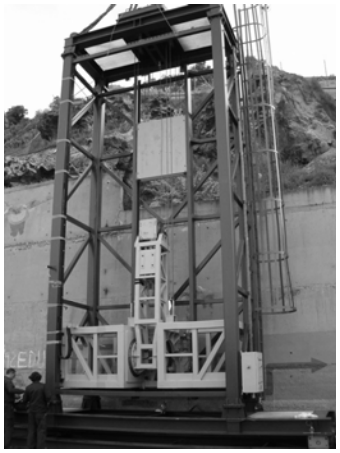

Figure 3. Primary Positioning System with both arm joint (yellow) and joint on tracks (green) of the EFTCoR system

The hazard H5 may imply the breakage of mechanical parts, the precipitation of components to the floor or damages to the human operator (R6, R7, R8). Taking into account the severity of the injury, the frequency of the exposure and the probability of avoidance, the RRC is R2A. Figure 4 shows the deployment partitioning of the system (using an extension of the standard UML notation), which accomplishes the R2A risk reduction factor for the hazard H5. The limitation of space in this paper does not allow us to give all the details related to the real implementation of the safeguard for this hazard (for instance, the concurrent state machines which we have used to model the concurrent monitoring of the system, etc.). Even so, the following description gives a good idea of the way in which the hazard has been considered:

- When a movement command is received, the *Man Machine Interface* (MMI) node forwards it simultaneously both to the *Control Unit* node (which will process it) and to the redundant node dedicated to monitor possible hazards (*Safety Control* node).

- The control node reads from a sensor the current position of the joint and controls directly the functioning of the motor. The safety control node will stop the motor when it detects a malfunction of the motor.
- Just before the execution of any command, the control node sends a message to the monitoring node authorizing the starting of the movement. From this instant, the control unit sends to the monitoring node, by means of pulses (*watch dog*), the current value just read from the sensor. The monitoring node answers this *tick* with an acknowledgement signal (*ack*) which includes as a parameter the estimated value of the motor position. Both nodes compute the curve of the discrete positions that must be reached by the robot arm. This timed discrete calculus is done by taking into consideration the initial value of the sensor and the command to be executed. Any difference between the calculated values implies an anomaly in the function of the robot movement. When a node detects a discrepancy in this value with respect to the estimation of the position values, an emergency signal is generated.

The different error conditions which could lead to a safe stopping of the robot in the previous example are the following: (1) Both the design and construction of the robot are done in such a way that, if a global fail of the system occurs then the robot will be mechanically fixed and returned to a safe mechanical state; (2) If any computing node does not work well (due to software or hardware errors) or the communication link fails then the other one will detect the discrepancy in the values. In this last case, it is essential that the control unit periodically reads the sensor data, although there is no current movement command in execution.

6. Related Works

One of the most popular approaches to identify, evaluate and manage safety requirements is the technique named *fault trees* [8]. These trees provide a graphical notation and a formal support that facilitate the analysis from the perspective of the system fails and its origins. However, they do not offer a global framework for requirement specification as a discipline. From the point of view of requirement refinement, our proposal is analogous to the use of fault trees; however, in our work the analysis of safety requirements is integrated and derived from the set of functional requirements of the system.

Letier et al [14] have proposed the use of KAOS for safety related requirement specification. They have introduced the concept of *obstacle* as *a set of non desirable behaviours*, the presence of those obstacles imply the obstruction in the fulfilment of the objective. At the same time, the negation of the obstacle generates the preconditions needed for the satisfaction of the requirements. The safety goals of the system being developed are formally specified by using temporal logic, and the obstacles (similar to hazards) are automatically obtained by the negation of the safety objectives and following the patterns given in their proposal [10]. However, the KAOS proposal does not provide a specific process for dealing with safety goals in the context of safety requirement specifications, nor does it consider factors such as severity, exposition time, etc, to be exploited during the analysis of the safety specification.

7. Conclusions

In tele-operated systems the specification of safety requirements is a major challenge due to both the combination of hardware/software components and the presence of potential injury to people or equipment. In this work, we have introduced a unified proposal for requirement specification in this domain, which provides the integration and derivation of safety requirements with the remainder. We have adopted a goal oriented approach due to the obvious benefits for analysis and evaluation of alternatives. Moreover, the formal base of the approach allows the verification of several properties (not mentioned for space limitations).

The consideration of very relevant standards and practical approaches (ANSI, Douglass and Leveson) for the domain of safety requirements has been the source to establish some methodological guidelines for safety requirement specification of tele-operated robotic systems. In these systems, it has been demonstrated that the use of catalogues and the perspective of traceability throughout the development process could improve the reuse of the knowledge based on a software product line approach [7]. In this paper, we have illustrated the application of these ideas in the scope of the EFTCoR research project.

Currently, we have two open issues: The first is related to the application of our proposal into other projects; and, the second is devoted to the extension of MetaEdit+ for the development of analysis tools, which help, for instance, to automatically determine the RRCs.

8. References

[1] ANSI/RIA R15.06-1999. *American National Standard for Industrial Robots and Robot Systems, Safety Requirements. Robotic Industries Association,* 1999.

[2] B. P. Douglass, *Real-Time Design Patterns. Robust Scalable Architecture for Real-Time Systems.* Reading, Addison-Wesley. 2003.

[3] EFTCoR. Requisitos del Sistema. Anexo I: Resumen de Requisitos Funcionales de los sistemas robóticas. Subproyecto ANCLA, Abril 2004. http://dynamica.dsic.upv.es:8090/

[4] EFTCOR: Environmental Friendly and cost-effective Technology for Coating Removal. (GROWTH G3RD-CT-00794) EU Project within the 5th Framework Program, 2003.

[5] C. Fernández, J.A. Pastor, P. Sánchez, B. Álvarez, A. Iborra: "Co-operative Robots for Hull Blasting in European Ship repair Industry. Robotics and Automation Magazine (RAM)", special issue on Industrial Robotics Applications, September 2005.

[6] E. Gamma, R. Helm, R. Johnson, J.Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, 2003.

[7] J. van Gurp, J. Bosch and M. Svahnberg "On the notion of variability in software product lines", Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA'01), 2001, pp 45—54.

[8] K.M. Hansen, A.P. Ravn, V. Stavridou, "From Safety analysis to software requirements", IEEE Transactions on Software Engineering, 24 (7), Jul 1998, 573-584.

[9] *ISO/IEC Standard 9126-1 SE- Product Quality-Part1: Quality Model*, ISO Co. Office, Geneva, June 2001.

[10] S. Kelly, K. Lyytinen, M. Rossi: "METAEDIT+ A fully configurable Multi-User and Multi-tool CASE and CAME Environment". Proceedings of 8th International Conference on Advances Information System Engineering, LNCS1080, Springer-Verlag, 1996, 1-21.

[11] A. van Lamsweerde and E. Letier, "Handling Obstacles in Goal-Oriented Requirements Engineering", IEEE Transactions on SE, Special Issue on Exception Handling, 26 (10), October 2000, 978-1005.

[12] R. Lemos, A. T. Saeed, "Analyzing Safety Requirements for Process-Control Systems", IEEE Software, 12(3), 42-53, May 1995.

[13] P. Letelier, E. Navarro, V. Anaya, "Customizing Traceability in a Software Development Process", 13th International Conference on Information System Development, Vilnius, Lithuania, September, 2004.

[14] E. Letier and A. van Lamsweerde, "High Assurance Requires Goal Orientation", Proceedings of International Workshop on Requirements for High Assurance Systems, Essen, Septembre 2002.

[15] N. Leveson. *Safeware: System Safety & Computers*, Addison-Wesley, 1995.

[16] E. Navarro, P. Letelier, I. Ramos, "Goals and Quality Characteristics: Separating Concerns", Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design Workshop, Oct. 25, 2004, Vancouver.