ELICERE: A Process for Defining Dependability Goals for Critical Computer Systems

Carlos H. N. Lahoz
Instituto de Aeronáutica e Espaço IAE
Praça Mal. Eduardo Gomes, 50
São José Campos-SP-Brasil 12228-904
Phone: 55 12 39474901
lahoz@iae.cta.br

João Batista Camargo Junior
Universidade de São Paulo POLI/USP
Av. Prof. Luciano Gualberto, trv 3, 158
São Paulo-SP-Brasil 05508-900
Phone: 55 11 30915401
joao.camargo@poli.usp.br

Abstract. This paper introduces the ELICERE process: a dependability requirement elicitation process applied to critical computer systems, based on a goal-oriented requirements engineering technique i* (pronounced *i-star*), and safety engineering techniques. After creating the system model using i* diagrams and their relationship components, such as goal, soft-goal, resource and task, these components are analyzed through guidewords based on HAZOP and FMEA. This process is applied to some critical elements of the system model, from where goals related to dependability are extracted. The authors believe that this interdisciplinary approach will allow promoting the identification of goals that meet the quality requirements related to safety, reliability and other dependability factors still in the project conception phase. Also, ELICERE intends to facilitate the communication between system and software engineering communities, during a project requirements engineering phase.

Introduction

Software plays a strategic role in critical computer systems, particularly in the aerospace project domain. More and more functionalities that were implemented by hardware are now implemented by software. The consequence is that software is also playing an increasing role in aerospace accidents. Leveson (Leveson, 2005) stated that the vast majority of software-related accidents, in the space project domain, were related to flawed requirements and misunderstanding about what the software should do. Problems in the requirements engineering activities, mainly in the elicitation activity, could contribute to produce poor, inadequate or even non-existent requirements. These problems can lead to mission losses, disasters, premature project termination or promote an organizational crisis. According to Hecht and Buettner (Hecht and Buettner, 2005), in the period from 1998 to 2000, nearly half of all observed spacecraft anomalies were related to software. The authors affirm that the generation of software requirements is the major source of errors in system development. They also state that a study of requirements-originated software failures showed that roughly half resulted from poorly written, ambiguous, unclear, and incorrect requirements. The rest came from requirements that were completely omitted. Most problems introduced into software can be traced directly to requirements flaws.

So, it is necessary that the requirements engineering techniques be improved to assure that the software contributes to the accomplishment of the system mission with safety and success.

The elicitation activity consists of the extraction and identification of the system software requirements. Good elicitation practices can avoid errors in identifying or understanding functional and interface requirements, which frequently lead to safety-related software errors

(Lutz, 1993). A manner of minimizing potential safety problems in critical computer systems, like those of space projects, is to introduce safety considerations during software requirements elicitation activity. This is a manner of create a common understanding of the dependability issues by the system and software engineering communities in the earlier projects phases.

Regarding the people involved in the requirements elicitation activities, actions should be implemented to obtain a larger interaction between systems engineers and software engineers, due to the fact that each of these communities have their own methods of capturing, specifying and managing requirements. As stated by Gonzáles (Gonzáles, 2005) while system engineers are learning to apply more methodological approaches to elicit and analyze requirements, software engineers are learning to take the broader view necessary to develop system solution.

Then, to integrate these two perspectives and to improve the communication between system and software engineers, the ELICERE process was created. ELICERE adopts the i* framework (Yu, 1995), for modeling the computer systems behavior and the guidewords based on HAZOP (HAZard and OPerability studies) (Ministry of Defense, 2000) (Redmill et al, 1999) and FMEA (Failure Modes and Effects Analysis) to extract goals related to dependability. After creating the system model through the i* strategic actor relationships and their goal, resource, task and soft-goal components, they are analyzed by guidewords proposed by ELICERE. ELICERE aims to create a dependability process suitable to use in computer systems projects developed by the Brazilian Institute of Aeronautics and Space (IAE), which is responsible for space projects like the Satellite Launcher, VLS-1.

First, this paper comments very briefly the ELICERE process and its main steps. Next, the i * diagrams and elements are described. The use of these diagrams was motivated by the easiness of representation of the system to be constructed: resources, tasks, goals and soft-goals, are a clear and easy way to be understood by heterogeneous working teams. Later, guidewords based on HAZOP and FMEA, and the specific questionnaire, suitable for the process, are introduced. The guidewords and questionnaire aims to elicit the potential risks of the system to be constructed and set a priority degree for the system elements that could receive dependability soft-goals. After, an example of the process application, in a set of elements related to the flight control of a hypothetical launching rocket, is presented. Finally, the main obstacles that an organization must overcome in elicitation activity are made, under the process, technical and cultural point of view.

The ELICERE Process

The ELICERE process addresses the requirement elicitation activity of a system project requirements engineering process (Kotonya and Sommerville, 2000). In general, this activity comprehends the establishment of the general business goals, an outline description of the problem to be solved and the system constraints identification.The focus of system engineering in this activity is to define goals or requirements related to the system functionalities and mission requirements, or what the system must do. The ELICERE purpose is to help the requirements engineer to define what the system cannot do, or what the system should do in order to minimize problems related to safety, security, reliability and so on. The process, through a systematic and cultural approach, can improve the product quality, mitigating problems such as ambiguity, risk behavior, unclear, besides omission of non-functional requirements.Figure 1 presents the ELICERE process main steps.

(1) Modeling the system with i*: the purpose of this activity is to create a system model, through the organizational requirements modeling technique i*. The modeling is used to represent a system in such a way that its hazards and vulnerabilities become evident, as well as what the mission goals are, what kind of elements the system needs to operate, and so on. The

constraints should be identified and represented by the five modeling elements of the i* technique: actor, resource, task, goal and soft-goal. In this activity two i* strategic models are used: Strategic Dependencies (SD) model and Strategic Rationales (SR) model. (2) Applying safety analysis: this activity applies ELICERE guidewords, to the i* components of the system modeled. This technique application will allow obtaining a characterization of the hazard evidences to be explored and investigated in order to improve the system dependability issues. To accomplish this activity it is necessary to apply a questionnaire to for each guideword to conduct the safety analysis that could result in a set of soft-goals for the system. This is an easy way to discover goals related to non-functional requirements (soft-goals) that allow minimize the preliminary system hazards.

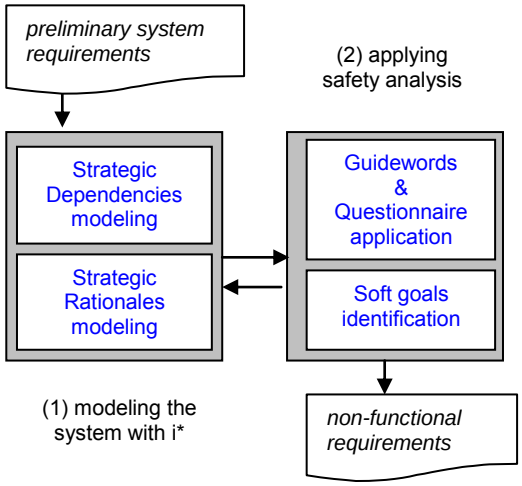

Figure 1. The dependability requirements elicitation process ELICERE

Through a set of simple steps and easy to use model representation and hazard guidewords, the ELICERE process intends to facilitate the communication between system and software engineers, during the requirements engineering phase of a project. The process looks for stimulating these engineers to discuss and find common dependability goals that can become non-functional system requirements.

The i* Modeling Technique

The first step of the ELICERE process is creating the system models, based on mission goals. The organizational system behavior description through goals, was proposed by several authors, who introduced techniques known as Goal-Oriented Requirements Engineering (GORE). GORE is based on the premise that, for design system based on computers, is necessary to consider the stakeholders profile, the business objectives, the plant behavior and the system as a whole, among others strategically questions for the organization. The i* technique was created from the PhD thesis of Professor Eric Yu, in the 1990‘s. This framework, which later gave origin to the Tropos project, is been applied in the Health Laboratory of the University of Namur, in Belgium, with the Arthur project (Petit and Rousseau, 2000); the Center of Scientific and Technological Research of the Culture Institute of Trento, Italy (ITC-IRST) (Perini and Susi, 2001); and several projects in Toronto University (Yu, 2006). Also, the researchers of the Human Computer Interaction Centers (HCI) at City University, created an innovative process using i* for specifying requirements for socio-technical systems, tailored to the air traffic control domain, called REquirements with SCenarios in User-Centred Engineering (RESCUE) (Maiden et al, 2003) (Maiden and Jones, 2004).

This framework is composed by two kinds of models: Strategic Dependency (SD) model and Strategic Rationale (SR) model. The SD model provides a description of dependency relationships among organizational actors, or system elements to be modeled. The actors, in this context, may be represented by both concrete and abstract entities, such as human being performing roles, electronic computational systems; sensors and actuators, intelligent agents, or any others entities, that is the specific approach, must represent the system components. This SD model has five types of components:

Actor – Active entities that carries out actions to achieve goals by exercising its know-how.
Goal – condition or state of the world that can be achieved or not.
Task – the activity that an actor or system must perform to reach some goal.
Resource – physical or informational objects in the world availability.
Soft Goal – goal related to subjective aspects of the system, which an actor should attend or meet.

The dependency relationship, Figure 2, is represented by the dependency link, which is a connection between two actors. This link indicates that one actor depends on another for something that is essential to the former actor for attaining a goal (Miden et al, 2006). The depending actor is called the ~~–d~~epender", the actor who is depended upon, the ~~–d~~ependee", and the process element around which the dependency, ~~–d~~ependum".

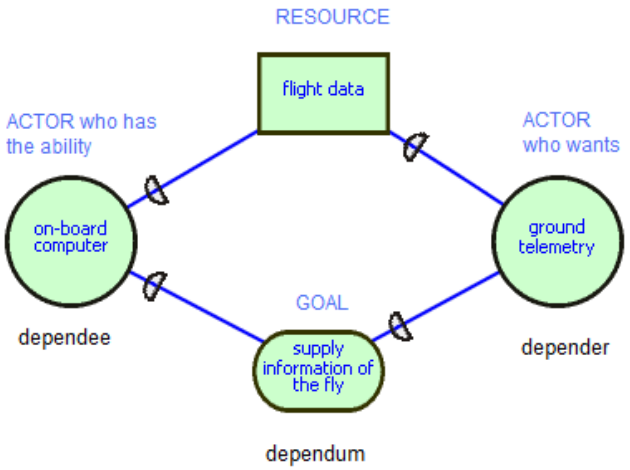

Figure 2. The i* Dependency Relationship Example.

The SR model is used to describe the internal behavior of the system actors, through their description in terms of relationships or rationales. These are activities or interests that should be accomplished, to represent the possible actor's behaviors which will be evaluated by the ELICERE process, regarding dependability. For systems with a high degree of automation, actors may be represented as electronic equipment, software functionalities, or subsystems.
Also, the SR model describes the tasks and resources necessary to actors to accomplish their goals. This model could be used to represent the (alternative) reasons behind the dependencies of several actors. Based on the same kind of SD components (actor, goal, task, resource and soft-goal), this model has incorporated three new types of relationship:

- Means-end link: suggests there could be other means of achieving the goal (alternatives).

- Task-decomposition link: describes what should be done in order to perform a specific task.
- Contributes-to soft goal link: a means-end link with a soft-goal as the end.

The ELICERE process uses the SD model to describe the external relationships among actors, and the SR model for detailing the internal behavior of an actor.
After making the SD model to obtain a strategic view of the system as a whole, and the SR model to observe the internal view and behavior of each actor of the system, the next step is to evaluate hazards and risks of its components for proposing soft goals related to dependability. A new approach for guidewords, based on safety analysis techniques as HAZOP and FMEA, was chosen in order to identify and prioritize the goals, soft goals, tasks and resources which should be evaluated for the system dependability improvement.

ELICERE Guidewords

The next step of ELICERE is to apply a safety analysis technique based on HAZOP and FMEA guidewords, addressed to the elements of the system modelled with i*.

The guidewords are used as a tool for the safety analysis conduction, aiding the hazard evaluation of the system components, anticipating their possible risks or failures. The ELICERE specific guidewords, that are strongly based on the HAZOP study nodes, will be applied in the goal, resource, task and soft goal of i* components, observing their dependency relationship.

These guidewords represent the deviation of design intent, taking into consideration mainly the i* components to be used in Programmable Electronic System (PES). Computer systems, communication systems, hardware devices (sensors and actuators), software or even human interface are some important actors considered during this step.

The HAZOP and FMEA originated a couple of approaches such as SHARD and LISA (Software Hazard Analysis and Resolution in Design/ Low-level Interaction Safety Analysis) (Pumfrey, 2000) and SFMEA (Software FMEA) that were used as reference for creating the ELICERE guidewords. While SHARD and LISA are more appropriated for hardware/software deviations, SFMEA is used to verify software requirements, specifically to analyze software requirements in space vehicles (Lutz and Woodhouse, 1997). The SHARD technique examines the information flow deviations, initiating with the output system or its functions. LISA examines events time deviations, such as interruptions, and physical resources used in the system operation. Pumfrey refined this guidewords based on a series of analyses of different computer systems and found that they were sufficient to prompt an analysis team to identify most (if not all) classes of failures (Conmy, 2005). The SFMEA uses forward searching to identify cause-effect relationship in which unexpected data or software behavior can result in failure modes. Lutz and Woodhouse (Lutz and Woodhouse, 1996) initiated their analysis focused on software, firstly observing, the effects of the failures and then identifying their causes. This approach refers to guidewords for failures related to software events and data flow events.

The ELICERE guidewords for i* resource take into account the deviation of design intent related to physical (equipment) or logical (flow of data) devices, that have a relationship with the actor analyzed. Table 1 shows the ELICERE guidewords that must be applied to the i* resource component and their correspondence in the SFMEA, SHARD/LISA and HAZOP guidewords.

Table 1. i* Resource Guidewords

SFMEA	SHARD/ LISA	HAZOP	ELICERE process
Absent Data	Omission Total	No	ABSENT RESOURCE
Incorrect Data	Commission	More Reverse	INCORRECT RESOURCE
	Omission Partial	Less Part Of	
	Value	Other Than	
Duplicate Data	Commission Repetition	As Well As	ADDITIONAL RESOURCE
Timing of Data Wrong	Early	Early Before	RESOURCE OUT OF TIME/ORDER
	Late	Late After	

ELICERE resource guidewords interpretations:

Absent resource: data not sent or received; sensor or actuators fail.

Incorrect resource: bad data; spurious signals; data incorrectly received; part of data received; inverted signal.

Additional resource: data saturation, redundant copies of data.

Resource out of time/order: data arrive too late/early to be used; Incorrect sequence of data/event.

The guidewords are associated to a questionnaire that allows identifying and classifying the possible project deviations. It also allows the classification of the deviation impact in the system behaviour, its severity degree and the potential risk of occurring. The idea is to select those system elements that have a high severity degree and more potential risk of occurring, in order to define soft-goals, or even goals, related to dependability. Hence, for choosing the relevant guidewords for a specific project deviation it was created a semantic connection between i*components and ELICERE guidewords. Each goal guideword has specific tasks and resources guidewords associated to it. A deviation of design intent related to a goal (goal guideword) has a set of deviation of design intent related to tasks and/or resources. The Figure 3 illustrates the guideword tree of a ―Goal is not achieved" and ―Goal is out of time/order" and their associated tasks and resources guidewords.

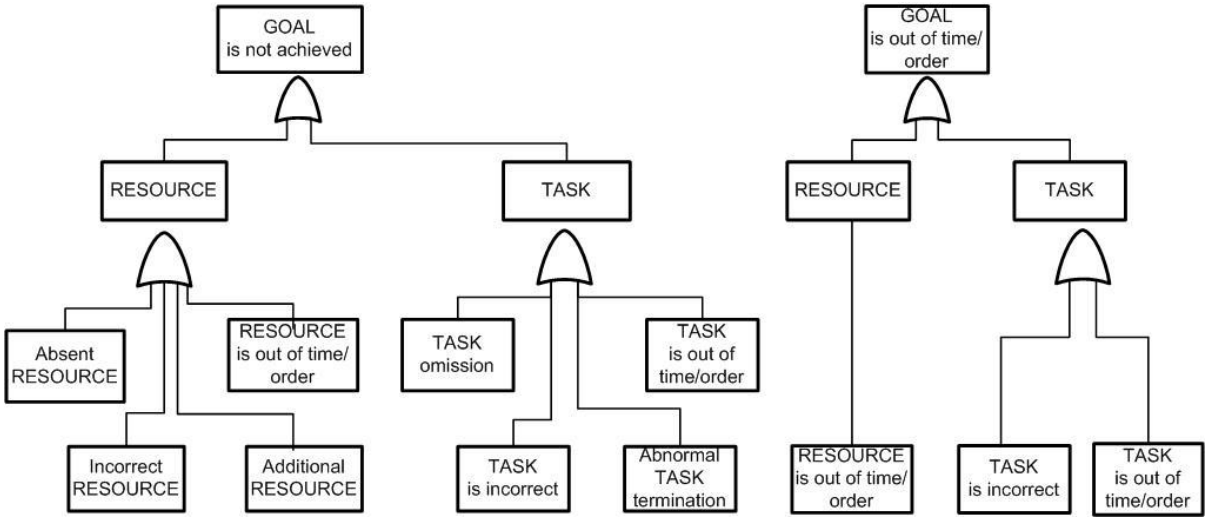

Figure 3. ELICERE tree of Goals guidewords.

The ELICERE questionnaire uses as reference the preliminary study proposed by Bush (Bush, 2005), applied to UK National Air Traffic Services - Departure Manager System (NATS-DMAN). Similarly, the Bush´s study uses the i* structures and semantics as the system model against hazard identification and analysis, based on HAZOP technique. The questions proposed by Bush, analyze the guidewords in order to indicate the failure mode, the worst effect, the severity risk level, the tolerable risk level and what soft goal should be added to ensure the system dependability. The severity risk degrees, as well as the tolerable risk level for ELICERE processes were based on the European Space Agency (ESA) (European Space Agency, 2002) and the safety regulations adopted by the Brazilian Space Agency (AEB) (Agência Espacial Brasileira, 2007).

The V-ALPHA case study

The ELICERE case study, presented herein, is based on a qualitative and descriptive single-case (Yin, 1984) (Zainal, 2008), and consists of a computer system project of a hypothetical launching rocket, called V-ALPHA. The idea is to use the case study results in the on-board computer system of the first rocket of Cruzeiro do Sul Program (Moraes et al, 2006).

For the sake of simplification this paper presents only the main components of the V-ALPHA control system. The first stage attitude control uses a movable nozzle system as actuators. The second stage uses only one movable nozzle and an auxiliary liquid thruster system for torque generation, called roll control system. Finally, the third stage presents a liquid thruster system that performs the maneuver for the satellite ejection.

The V-ALPHA digital controller, which includes the on-board computer and its embedded software, is responsible for performing in real time algorithms and control loops needed to conduct the vehicle in its nominal trajectory, to command the sequence of flight events, and to decide what the best orbit is for ejecting the satellite. The input information comes from the inertial measure unit that supplies information for the several algorithms of the embedded software to calculate inertial position, velocity and to calculate the instantaneous specific force to be used in the sequence of events of the vehicle (Leite Filho, 1999).

During the flight, the vehicle must be capable of detecting and processing a sequence of events, starting this process in the launching pad (five seconds before lift-off) - when the inertial measure unit begins its operation - and finishes with the end of the ballistic phase, that precedes the third stage spin-up. The main goal of the digital controller is to execute the tasks of navigation, attitude control, roll and pitch over control, guidance and the execution of sequence of flight events in order to assure that the vehicle fulfills its mission, which is ejecting the satellite in a desired and possible orbit.

Modeling the V-ALPHA with i*. The first ELICERE process step is to create the SD and SR models of the V-ALPHA digital controller. The actors selected for this case study were those that represent the flight control system components of the vehicle, under the digital controller point of view, such as sensor, actuators and the tasks responsible for performing the mission objectives.

The SD diagram describes a high level control system model, through a representation of the system actors and their external behavior and dependency relationships. The digital controller (**DC**) actor receives information from the inertial measure unit (**IMU**) to thus stabilize and control the vehicle. **IMU** is able to identify the variations of the vehicle inertial reference coordinates (**X, Y, Z coordinates**) and to supply this data to **DC**. During the rocket flight, **DC** must execute several algorithms using the **IMU** data to calculate the position and the angular

velocity of the vehicle (navigation), to determine the attitude profile that the vehicle should accomplish (guidance), to calculate when the pitch over movement should be initiated and determine the instant when the third stage should be already be pointed in the correct direction.

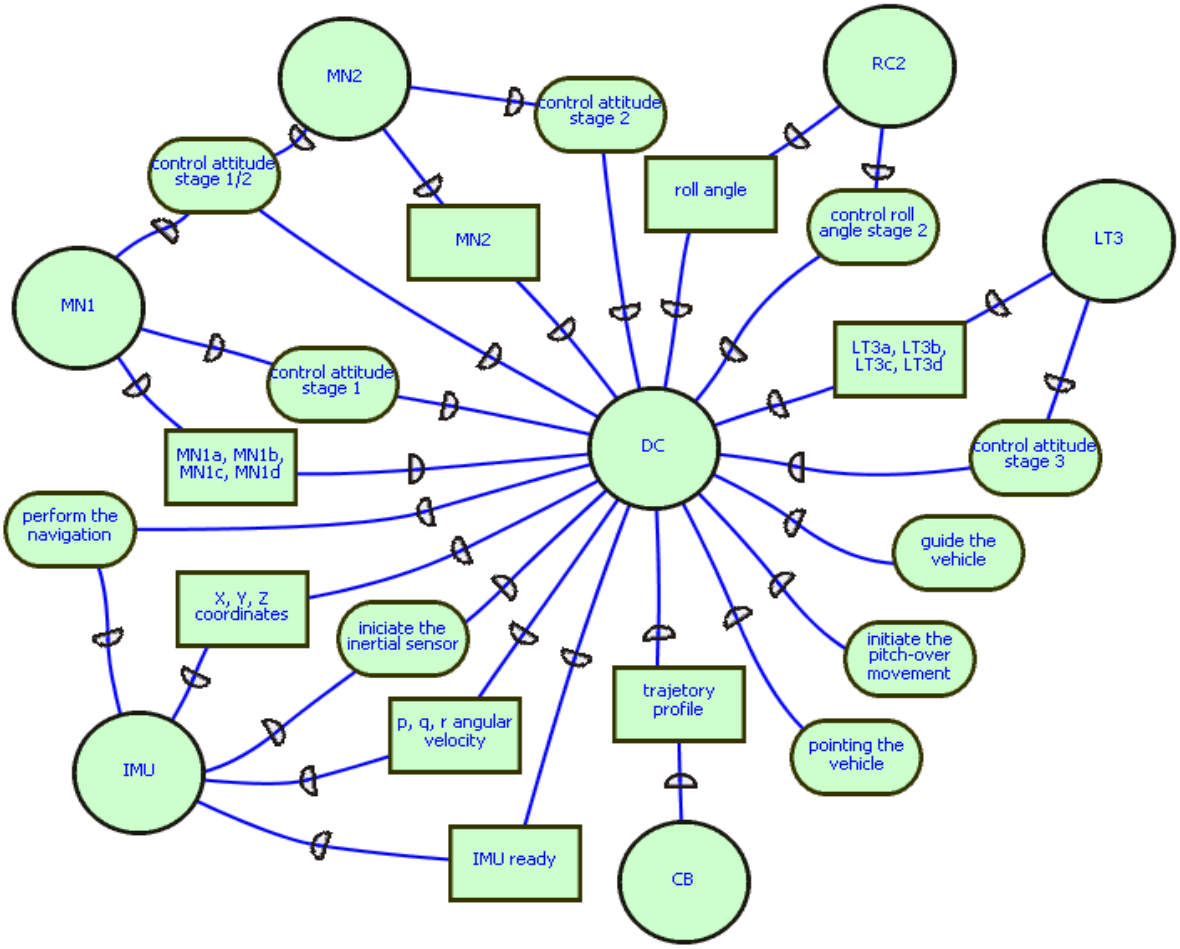

Figure 4. The V-ALPHA SD model of DC goals and resources.

The control actuators are necessary for the **DC** to guide and control the vehicle during the whole flight: four movable nozzles in the first stage of the flight (**MN1**), one movable nozzle (**MN2**) and the roll control system (**RC2**) in the second stage, **MN1**, **MN2** and **RC2** when the first and second stage are working simultaneously, and finally one liquid thruster system (**LT3**) in the third stage. Also, the **DC** actor follows a trajectory profile, with several parameters and inputs, loaded by the control bench (**CB**) few minutes before the lift-off, when the vehicle is ready to launch.
In Figure 4, a SD model represents a subset of goals and resources and their dependency relationship among **DC**, the **IMU** sensor and the **MN1**, **MN2**, **RC2**, **LT3** actuators.

The SR model describes the internal behavior of an actor, also presenting possible alternatives to attain their goals. Figure 5 shows an example of how to represent an SR model for V-ALPHA digital controller. The actor **DC** needs to **control attitude stage 1** goal so as to accomplish its mission. To meet this goal, the task **Control Movable Nozzle First Stage** was created, that was decomposed into several sub-tasks, such as **Calculate Position Error**, **Calculate Pith and Yaw Rates**, **Calculate Roll Rate** and **Select Actuation System**.

The SR model represents the connections (dependency) and interdependencies between one and another SR model of the V-ALPHA system. With this view, it is possible to affirm that the

MN1 actor has a critical dependency (represented by **X**) on the **DC** actor, which supplies the resource **MN1a**, **MN1b**, **MN1c** and **MN1d** (four movables nozzles angles) needed to accomplish the goal.

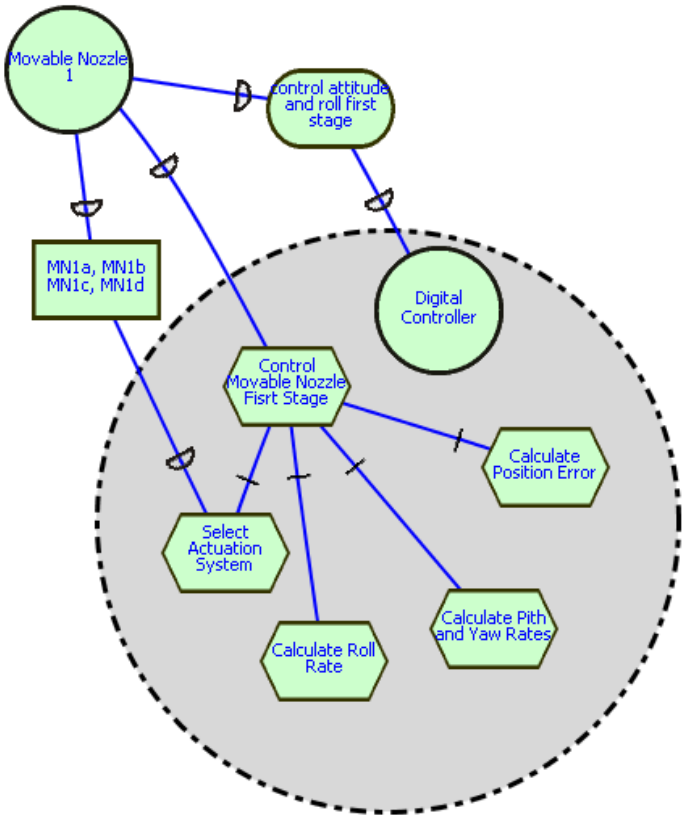

Figure 5. SR diagram: Task Control Nozzle First Stage of the DC actor.

After building the SD model to obtain system actors external behavior view, and the SR model to observe the each actor internal behavior view, the next step is to evaluate hazards and risks of the V-ALPHA components to propose soft goals related to dependability.

Applying Safety Analysis. The risk associated with the non satisfaction of the dependence relationship of the V-ALPHA components should be verified by the guideword analysis. Applying safety analysis in the SR model, Figure 4, the impact of the lack of the resource used in the dependence relationship between **DC** (dependee actor) and **MN1** (depender actor) was analyzed. The **Control Movable Nozzle First Stage** task is created to meet the **control attitude stage 1** goal and this task cyclically supplied the four first stage movable nozzles angles, represented by the resource **MN1a**, **MN1b**, **MN1c** and **MN1d**.
For the **Control Movable Nozzle First Stage** task dependence relationship with DC and MN1 actors the task guidewords interpretations are:

Abnormal Task Termination means that a part of the task did not perform correctly until the end.
Task Omission means that the task was not started/or performed to satisfy and accomplish the **control attitude stage 1** goal.
Task is Incorrect means that an incorrect logic or event did not allow this task to perform correctly.
Task is Out of Time/Order means that the task is not executed in time to satisfy the **control attitude stage 1** goal.

The soft-goals obtained from the questionnaire answers aims to create arguments to justify and prioritize the set of dependability goals identified. Figure 6 and 7 present a report example of the **Task Omission** and **Task is Out of Time/Order** guideword interpretation for the **Control Movable Nozzle First Stage** task.

TASK **Control Movable Nozzle First Stage**

GUIDEWORD: **TASK OMISSION**:

Guideword interpretation: The task was not started and executed in order to satisfy and accomplish the goal.

What is the TYPE OF FAILURE? Software.

What is the FAILURE MODE? The task failed to supply the angles of Pitch, Yaw and Roll of the first stage attitude control.

What is the worst FAILURE EFFECT? LOSS OF MISSION. The lack of the task does not allow the correct angles to be selected for the first stage movable nozzle, compromising the entire mission.

What is the SEVERITY OF FAILURE? Critical (2).

What is the TOLERABLE RISK? Low Risk (1).

What are the Soft-Goals suggested?

1) Verify before its execution if the task is created.

2) Assure that the task is executed with the correct priority.

Figure 6. Task Omission guideword analysis report.

TASK**: Control Movable Nozzle First Stage**

GUIDEWORD: **TASK OUT OF TIME/ORDER:**

Guideword interpretation: The task was started and executed out of time to satisfy and accomplish the goal.

What is the TYPE OF FAILURE? Software.

What is the FAILURE MODE? The task failed to supply in time, the angles of Pith, Yaw and Roll of the first stage attitude control.

What is the worst FAILURE EFFECT? PERFORMANCE DEGRADATION. The delay of execution the task doesn't allow the correction of the vehicle trajectory in the correct time.

What is the SEVERITY OF FAILURE? Critical (2)

What is the TOLERABLE RISK? Low Risk (1)

What are the Soft-Goals suggested?

1) Assure the temporal behavior of the task.

2) Verify the temporal properties of the system.

Figure 7. Task is Out of Time/Order guideword analysis report.

The soft-goals stated by the Figure 6 are related to consistency and schedulability non-functional requirements. They emphasize the necessity of including some software mechanisms to assure that the task is created and executed when it is called by the system. With respect to the Figure 7, soft-goals stated are related to the response time non-functional requirement. The main objective is to guarantee that the task temporal behaviour is kept normal. These soft-goals aims to identify what are the appropriate non-functional requirements that should be defined to assure that the V-ALPHA Control Movable Nozzle First Stage task is dependable.

Final Considerations

The limitations and barriers that the elicitation activity has to overcome are not small, and frequently they need to be dealt with a cultural change in an organization, mainly regarding to safety culture. In space system development, problems involving inadequate safety practices, poor requirements specifications and insufficient knowledge of the system to be developed have been already reported several times. The European Space Agency (ESA) initiative (Hjortnaes, 2003) about the analysis of the results from Technical Reviews of more than eighteen projects, as well as Leveson studies (Leveson, 2004), raised safety and requirements problems related to software, that could lead to a poor understanding of the system context or its operational environment. The official report (Serviço Público Federal, 2004) of the VLS-1 third prototype official accident investigation argues that an appropriate _safety culture‘ in the project received relatively little attention. Based on this accident investigation report, Almeida and Johnson (Almeida and Johnson, 2005) stated that some areas of the Brazilian space program had not recognized the importance of safety management systems.

Taking into account these several space projects, even when the organizations involved are highly mature space agencies, there are many problems related to requirements and safety culture. Johnson's incident investigation about the Mars Climate Orbiter (Johnson, 2003) identified some barriers related to technology, process and people point of view that must be overcome to mitigate the lack or poverty of defining requirements related to dependability in critical computer systems.

From a technology point of view, it is possible to observe inadequate or non-existent use of elicitation technique for representation the mission goals. Interviewing, prototyping or even the document studies are often misused, offering not completely reliable results. In addition, an inappropriate model representation of the system components and their interactions may not show the real interdependence between actors. Hence, the causes for failures of software systems are invariably related with incorrect interactions between components (de Lemos, 2002). A poor representation of the system and its subsystems interactions may not appear until very advanced phases of a system development, causing financial loss and problems with project schedules.
Critical computer systems, like those of a space rocket, have heterogeneous development teams, need to have their models represented in a common language in order to avoid or minimize most of interpretation problems. ELICERE process uses the i* diagrams to represent the system external behavior and the internal subsystem behavior, creating a common understanding for system and software engineers. ELICERE apply easy to use guidewords and a questionnaire about the system components (actors), mainly the components that have strategic interactions.

From a process point of view, lack or poor requirements process model, as well as precarious safety approach, are strongly related to immature team and organization. Many times, for reasons external to a project, such as non-availability of qualified personnel, inefficient organizational structure or time and resource constraints, requirements elicitation activities are neglected in favor of others, like the ones directly related to code development. Even when an elicitation activity is established, a partial process monitoring makes impossible the follow-up of changes that occurs during the requirement life cycle. Another frequent obstacle in requirement process is inadequate communication between stakeholders. Inefficient communication during the elicitation process may make impossible the requirements identification and consistency, so that they remain unstable during project life. The ELICERE proposes to join system engineers and software engineers in a single requirement elicitation

process, helping the definition of system and software goals related to dependability. Through very simple steps, artifacts and roles, the ELICERE creates a more propitious environment for identification and negotiation dependability goals.

From a people point of view, different system and software designers cultural approaches may become an impassable obstacle. Software is a relatively new discipline, according to systems engineers, and many times the participation of the software team in the system requirement elicitation activity is viewed as not important. Another critical factor is the limited understanding of the problem domain and the stakeholders partial view, that negatively impacts any requirements elicitation initiative. Without a clear and precise understanding of the problem domain it is difficult to interact with stakeholders and go beyond the individual system points-of-view. Top management must balance interests, determine priorities and foster stakeholders consensus to introduce non-functional requirements in the project. Without such a balance, opposing currents of seemingly common interests may slowly and gradually undermine a project. ELICERE uses the questionnaire results to evaluate the dependability of each system components (goal, resource and task) and supply to the system and software engineering arguments to discuss with project manager the importance of soft-goals.

ELICERE puts together in a same process these two main issues: requirements and safety for system and software engineers to discuss and discover new dependability goals. A clear and precise understanding of the problem domain becomes easier the interaction of the people involved and goes beyond the individual views of the system.

References

Agência Espacial Brasileira 2007 AEB - Regulamento Técnico Geral da Segurança Espacial, Brasília, Brasil, 21 March

Almeida, I M. and C. W. Johnson 2005 Extending the Borders of Accident Investigation: Applying Novel Analysis Techniques to the Loss of the Brazilian Space Programme's Launch Vehicle-VLS-1-V03",http://www.dcs.gla.ac.uk/~johnson/papers/papers.html

Serviço Público Federal 2004. Ministério da Defesa, Comando da Aeronáutica, Departamento de Pesquisas e Desenvolvimento, –Relatório da Investigação do acidente ocorrido com o VLS1-V03, em 22 de agosto de 2003, em Alcântara, Maranhão", São José dos Campos, Fev http://www.aeb.gov.br

Bush, D. 2005 Modeling Support for Early Identification of Safety Requirements: A Preliminary Investigation, 4th International Workshop on Requirements for High Assurance Systems -13th IEEE International Requirements Engineering Conference, Paris, France, August 29 – September 2

Conmy, P. M. 2005 Safety Analysis of Computer Resource Management Software, PhD. thesis, University of York

de Lemos; R. 2002 Analysing failure behaviors in component interaction, *The Journal of System and Software*, August

European Space Agency 2002 ESA -European Cooperation for Space Standardization, Space Product Assurance – Dependability, ECSS-Q30B, ESA-ESTEC, Noordwijk, The Netherlands, 8 March 2002.

Gonzáles, R. 2005 Developing the Requirements Discipline: Software vs. Systems, *IEEE Software*, March/April 2005, pp 59-61.

Hecht, M. and D. Buettner 2005 Software Testing in Space Programs, *Crosslink*, Volume 6, Number 3 (Fall) http://www.aero.org/publications/crosslink/fall2005/06.html

Hjortnaes, K. 2003 ESA –Software Initiative, ESA Report", May `http://klabs.org/richcontent/software_content/presentations/esa_software_initiative_final_may_7.pdf`

Johnson, C. W. 2003 A Handbook of Accident and Incident Reporting, Glasgow University Press, Glasgow

Kotonya, G. and I. Sommerville 2000 Requirements Engineering,. John Wiley & Son Ltd, West Sussex, UK

Lauesen, S. 2002 Software Requirements: Styles & Techniques", Person Education

Leite Filho, W. C. 1999 Control System of Brazilian Launcher, 4th ESA International Conference on Spacecraft Guidance, Navigation and Control System, ESTEC, Noordwijk, The Netherdlands, October

Leveson, N. G. 2004 The Role of Software in Spacecraft Accidents, *AIAA Journal of Spacecraft and Rockets*, Vol. 41, No. 4, July

———. 2005 A New Approach to System Safety Engineering, Massachusetts Institute of Technology, Draft of New Book `http://sunnyday.mit.edu/book2.pdf`

Lutz, R. R. 1993 Analyzing Software Requirements Errors in Safety-Critical, IEEE International Symposium of Requirements Engineering

Lutz, R. R. and R. M. Woodhouse 1996 Experience Report: Contributions of SFMEA to Requirements Analysis 2nd IEEE International Conference on Requirements Engineering, Colorado Springs, USA, April 15-16

———. 1997 Requirements Analysis Forward and Backward Search Annals of Software Engineering - Special Volume on Requirements Engineering, (3), v.3 pp. 459-475.

Maiden, N. A. M. and S. V. Jones 2004 Dependability in RESCUE: A Concurrent Engineering Approach to the Specification of Requirements for Air Traffic Management Dependability, Systems and Networks workshop on interdisciplinary approaches, Florence, Italy, June 28 - July 1

Maiden, N. A. M., N. Kamdar and D. Bush 2006 Analyzing i* System Models for Dependability Properties: The Uberlingen Accident, 12th Working Conference on Requirements Engineering: Foundation for Software Quality, Luxembourg, Grand-Duchy of Luxembourg, June 5-6

Maiden, N. A. M., S. V. Jones and M. Flynn 2003 Innovative Requirements Engineering Applied to ATM, Proceedings ATM - Air Traffic Management, Budapest, Hungary June 23-27

Ministry of Defense 2000 MoD - HAZOP Studies on Systems Containing Programmable Electronics, Part 1 Requirements, Def Stan 00-58, Draft Issue 2, 2000.

Moraes, P. Jr., D. S. Carrijo,A. Garcia, L. E. L. Costa, U. C. Oliveira, A. Santana Jr., D. J. F. Villas Boas and M. K.Yamamoto 2006 An Overview of the Brazilian Vehicle Program Cruzeiro do Sul, 57th Intenational Astronautical Congress, Valencia

Perini, A. and A. Susi 2001 Modeling Information Technology Needs in the Agriculture Domain, Tropos Workshop, Trento, Italy, November 15-16

Petit, M. and A. Rousseau 2001 Using i* for early requirements of a computer system for hospital emergency departments, Tropos Workshop, Trento, Italy, November 15-16

Pumfrey, D. J. 2000 The Principled Design of Computer System Safety Analyses, PhD. thesis, University of York

Redmill, F., M. Chudleigh, M.and J. Catmur 1999 System Safety: HAZOP e Software HAZOP, John WILEY, West Sussex, England

Yin, R. K. 1984 Case Study Research: Design and Methods, Beverly Hills, Calif: Sage Publications, 1984.

Yu, E. 1995 Modeling Strategic Relationship for Process Reengineering, PhD. thesis, Toronto University.

———. 2006 Social Modeling for Requirements Engineering: Why, Where, and How, Workshop on Requirements Engineering, Rio de Janeiro, Brazil, July 13-14.

Zainal, Z. 2008 Case study as a research method, *Jurnal Kemanusiaan* bil.9, Jun 2007. Link: http://www.fppsm.utm.my/jurnal/JK9D06/JK9_WANKHAIRUZZAMANWANISMAIL.pdf

BIOGRAPHY

Carlos H. N. Lahoz works in Aeronautics and Space Institute (IAE) in Brazil, where coordinates two projects: the on-board software and the control bench software for the Brazilian Satellite Launch Vehicle (VLS). He is a PhD student in safety and dependability area at Escola Politécnica da Universidade de São Paulo (POLI/USP). Teaching in Computer Science graduate course at Universidade Paulista (UNIP). He is a Assistant Teacher since 2003. During 2005-2006 was Invited Teacher in the Aerospace Master Engineering course at Instituto Tecnológico da Aeronáutica (ITA).

João Batista Camargo Junior works in Escola Politécnica da Universidade de São Paulo. Since 1988 João Batista is an associated professor of POLI-USP. He is a PhD in computer safety systems analysis and leader of the Safety Analysis Group (GAS) in Computer and Digital System Engineering Department. His research interests are computer safety techniques and safety analysis methodology.