Analysing Information Integrity Requirements in Safety Critical Systems

Mohamad Gharib and Paolo Giorgini

University of Trento - DISI, 38123, Povo, Trento, Italy
{gharib,paolo.giorgini}@disi.unitn.it

Abstract. Organizations' assets are subject to different threats; which are addressed, usually, by different security solutions. Nonetheless *i** modeling language was not developed with security in mind, which motivated the development of other languages (e.g., SI*) that focused on capturing the security requirements (e.g., privacy) of the system-to-be, but far less attention has been paid for capturing information integrity requirements. Capturing information integrity requirements represents an important need for safety critical systems, where depending on incorrect or inconsistent information may lead to disasters and loss of humans' lives (e.g., Air Traffic Control Management Systems). In this paper we present a novel methodology for developing safety critical systems that extends *i*/* SI* modeling languages with the required concepts and primitives for modeling and analyzing the requirements of safety critical systems, with a special emphasis on information integrity requirements.

1 Introduction

Organizations' assets, especially information, are subject to different kinds of threats. Usually, these threats are captured, prevented or mitigated by different security solutions. The last few years have seen growing efforts for integrating security into the early system development process, since it is the best way to deal with the organizational requirements. For instance, SI* [3] offers a conceptual framework for modeling and analyzing security requirements (mainly privacy and confidentiality) starting from the organizational setting of the system-to-be.

Nowadays, we are more and more experiencing complex systems that are not simply composed by technical components but where organizations, people, and processes become integral part of the system itself. Considering only the technical aspects of the system leaves human, social, and organizational aspects outside the system's boundary and then opening to vulnerabilities that may arise at business and organizational level. For example, in an ATM system, the *ground controller*, based on its role, is not allowed to issue any taxiing information concerning active runways, unless he was permitted by the *local controller*. This can be captured only by the analyzing the system organizational aspects. Furthermore, *Air Traffic Controller Officer (ATCOs)* depends on the *captain* to report the airplane's position when the airplane is out of the radar coverage

area, i.e., *ATCOs trusts* that the *captain* will keep updating (modifying) the location information as required, which prevents inconsistent information within the system that might lead to disasters. Similarly, *trust* can be only captured by the social aspects of the system.

To this end, we advocate that any solution for information integrity related problems should consider the social, organizational and the technical aspects of the system. In this work, we propose a novel methodology for developing safety critical systems that extends *i*/* SI* modeling languages [2,3] with the required concepts and primitives to capture the requirements of safety critical systems with a special emphasis on information integrity requirements. The rest of the paper describes the research objectives, scientific contribution, and finally we outline our conclusions, and discuss the ongoing and future work.

2 Objectives of the research

The main objective of this research is to provide a requirements engineering methodology for developing safety critical systems. In particular, it will provide the following contributions: 1- a modeling language for designing safety critical systems that extends *i*/* SI* modeling languages with the required concepts and constructs for capturing the requirements of such system (especially information integrity requirements); 2- a requirements engineering methodology, that allows for the systematic design of safety critical systems, it aim to support all activities related to requirements analysis. 3- a formal framework to support designers in the requirements verification and validation; and 4- CASE tool to assist designers during the system development process.

3 Scientific contributions

We introduce the new concepts in section (3.1), and the methodology in (3.2).

3.1 The extended modeling concepts

Our modeling language extends the *i*/* SI* modeling languages with several concepts to capture information integrity requirements, including critical information, critical goal, information producer and consumer, and information integrity provision. The first is used to determine which information is critical to the system performance and its integrity has to be preserved, while the second is used to represent the stakeholders' critical objectives, and it is used to determine where information integrity requirements are needed. The third is used to define the initial sources of information and the actor who has full modification permission control over information it produces, while information consumer is used to determine if the integrity of information has been preserved at its final destination. The last is used to represent information provision that is able to preserve the integrity of the provided information. Furthermore, we refine the notions of delegation by introducing the delegation degree based on the trust degree. In the following sections, we define each of these concepts.

Information and information integrity Information can be produced by *information producers* (represent the initial source of information) in several different ways. For instance, information can be *generated* internally or *acquired* from physical objects (e.g., *ATCOs* is able to *acquire* information about airplane location by depending on the radar). Moreover, we call the actor(s) who consume(s) information as *information consumer(s)*, i.e., information is within the objectives of information consumer(s). In the case of safety critical systems depending on incorrect or inconsistent information to perform some activities is not acceptable since it might produce disaster. Thus, preserving information integrity (information integrity requirements) represents an important need for such systems. In this work, information integrity can be evaluated by 3 dimensions, namely, accuracy, completeness and consistency. While information integrity requirements mean preserving these 3 dimensions.

Critical information and critical goals It is well known that not all goals have the same criticality to the organization's performance. Thus, we introduce the *critical goal* concept, which is used to represent the stakeholders' critical objectives, and can be described as any goal that its failure might results in major problems to the organization, i.e., such goals should never fail.

Currently, we define two reasons that might threaten the satisfaction of critical goal: 1- consuming incorrect or inconsistent information; 2- problems that may rise and negatively effects the satisfaction of the goal. The first reason can be avoided by preserving the integrity of information consumed by the critical goal. To this end, we call such information as *critical information* that its integrity should be preserved at any given time. While the second reason can be avoided if all problems that might arise and negatively affect the critical goal satisfaction were detected [1] and solved before its occurrence. For example, *ATCOs* should "manage the airplanes traffic safely" (critical goal) that is why the integrity of information consumed by this goal should be preserved. Furthermore, if *ATCOs detects* that there will be an air traffic increase in his sector, and "manage the air traffic safely" is threatened, he should take some actions to *solve* this *problem* by avoiding its occurrence (e.g., delay or change the path of some flights).

Delegation and trust An actor might not have the capabilities to fulfill his objectives (goals). Thus, it delegates them to other actors. SI* introduces the notion of goal delegation, which identifies the transfer of responsibilities concerning a goal satisfaction among actors, where an actor (delegator) delegates the achievement of a goal (delegatum) to another actor (delegatee). In this work, we proposed a refinement of goal delegation by introducing the notions of delegation and trust. Moreover, we introduce 4 different degrees of trust [full, partial, limited, no] trust. Consider for example, an *ATCOs* delegates the goal "manage safely separation with other planes" to the *airplane captain*, the trust between *ATCOs* and the airplane captain concerning the goal satisfaction, will be eval-

[1] Certain information is used to detect the expected occurrence of each problem

uated based on the *airplane captain's* capabilities, problems detecting [2] and solving.

Full trust: *ATCOs* has a full trust that the *airplane captain* is able to satisfy the goal "manage safely separation with other planes", if and only if, the captain is able to satisfy the goal, and he is able to detect and solve all problems that may rise during the goal satisfaction.

Partial trust: *ATCOs* has a partial trust that the *airplane captain* is able to satisfy the goal "manage safely separation with other planes", if the captain is able to satisfy the goal, but he is not able to detect all the problems that may rise during the goal satisfaction, even he is able to solve them.

Limited trust: *ATCOs* has a limited trust that the *airplane captain* is able to satisfy the goal "manage safely separation with other planes", if the captain is able to satisfy the goal, but he is neither able to detect nor solve all the problems that may rise during the goal satisfaction.

No trust: *ATCOs* has no trust that the *airplane captain* is able to satisfy the goal "manage safely separation with other planes", if, simply, the captain is not able to satisfy the goal.

Furthermore, we extend the notion of goal delegation introduced in the previous languages (e.g., SI*) based on the different degrees of trust. We introduce three types of goal delegation: 1- full delegation; 2- partial delegation; and 3- limited delegation, they can be defined as follows:

Full delegation: The delegator fully trusts that the delegated goal will be satisfied, since the delegatee is able to satisfy the goal, and it is able to detect and solve any problem that may arise during the goal satisfaction.

Partial delegation: The delegator partially trusts that the delegated goal will be satisfied, since the delegatee is able to satisfy the goal, but it is not able to detect all problems that might rise during the goal satisfaction even if it is able to solve them.

Limited delegation: The delegator limitedly trusts that the delegated goal will be satisfied, since the delegatee is able to satisfy the goal, but it is neither able to detect nor solve problems that might rise during the goal satisfaction.

Ex1. *ATCOs fully delegates* "manage safely separation with other planes" to *captains*, if they were flying under Visual flight rules (VFR), where VFR require a *captain* to be able to control the airplane's attitude, navigate, and avoid obstacles by itself, i.e., they are able to detect and solve all the problems that may rise during the satisfaction of the goal.

Ex2. *ATCOs partially delegates* "manage safely separation with other planes" to *captains* at airways intersections, since *captains* are not able to detect if the intersection is being used by others, even they are able to solve such problem.

[2] The integrity of information used to detect the expected occurrence of each problem has to be preserved, especially in the case of goal delegation, since it is not necessarily that the actor who detect the problem is the same actor who supposed to solve it

Ex3. *ATCOs limitedly delegate* "airplane safely landing" to *airplane captains*, since *captains* are not able to neither detect nor solve any unexpected problem that might rise during the satisfaction of this goal.

In this case of partial and limited delegation the goal satisfaction is threatened, since the delegator does not have a full trust that the delegatee can satisfy the goal. It is well known that the lack of trust can be tolerated with monitoring the delegatee performance. In this work, we introduce two types of monitoring:

Partial monitoring: the monitoring actor knows all the possible ways in which the achievement of the goal can be performed by the monitored actor, i.e., the monitoring actor is able to detect if the monitored actor is not performing the achievement of the goal by one of these ways.

Full monitoring: the monitoring actor knows exactly how the achievement of the goal should be performed by the monitored actor, i.e., the monitoring actor is able to detect if the monitored actor is not performing the achievement of the goal as planned at any moment.

Ex4. In **Ex2** *ATCOs* should *partially monitor* the delegated goal satisfaction, since only the *ATCOs* is able to *detect* such problems, and asks the *captain* to alter his flight path when it is needed.

Ex5. In **Ex3** the *ATCOs* should *fully monitor* the delegated goal satisfaction, since *captain* is not able to neither detect nor solve any unexpected problems that may rise during the satisfaction of this goal.

3.2 The requirements engineering methodology

The methodology aims for the systematic design of safety critical systems, it is intended to support all activities related to requirements analysis process, including the requirements verification and validation process to determine whether the model satisfies the stakeholders' requirements, and to ensure that the requirements are correct, complete, and consistent [3]. The process starts with the actor modeling, in which actors are modeled along with their objectives, entitlements, and capabilities. Then, critical goals are defined, goals are analyzed and refined, and the criticality is propagated in the case that the critical goal. Based on the criticality of the consuming goal critical information is determined. Goal delegation, information [integrity] provision are modeled. Furthermore, social modeling starts with trust modeling, and based on the trust degree the delegation types are determined and modeled. Finally, based on the goal delegation type, full or partial monitoring are added. The methodology, including the new concepts, notations and the tools, will be evaluated using an ATM case study. Figure 1 shows the main phases of the requirements analysis process.

[3] Not included in this paper, some of the concepts are formalized in [1]

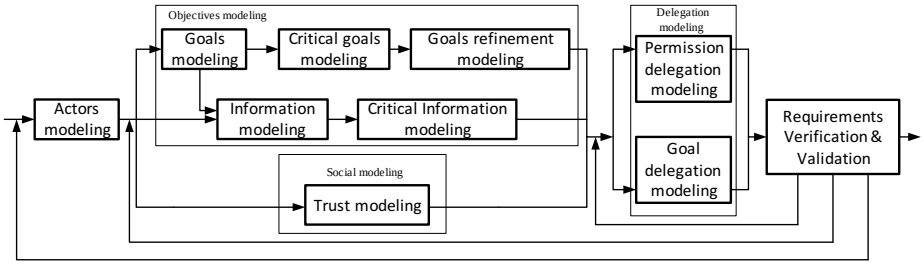

Fig. 1. Requirements Analysis Process

4 Conclusions

In this paper, we extended *i */* SI* with several concepts for capturing the requirements of safety critical systems, we focus more on information integrity requirements, and refine the notions of goal delegation based on the different degrees of trust. Furthermore, we showed how the requirements of the system-to-be will be constructed by the methodology.

5 Ongoing and future work

We are considering the following topics for the future work:

- The modeling language will be extended for capturing the related information integrity dimensions (accuracy, completeness and consistency).
- The methodology will be extended to capture the requirements of business critical systems beside the safety critical systems.
- We intend to increase the number of the design properties that our model is able to check (currently we have 8).
- A CASE-Tool that allows designers to verify the correctness of the model will be developed.

Acknowledgments The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant no 257930 (Aniketos) and 256980 (NESSoS).

References

1. Mohamad Gharib and Paolo Giorgini. Analysing information integrity requirements in safety critical systems. *The 3rd International Workshop on Information Systems Security Engineering WISSE13. To appear.*, 2013.
2. Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering.* Ph.d. thesis, Toronto, Ont., Canada, Canada, 1996. AAINN02887.
3. N. Zannone. *A requirements engineering methodology for trust, security, and privacy.* PhD thesis, PhD thesis, University of Trento, 2006.