Risk in Secure and Dependable System: a Survey

[1]Yudistira Asnar, [1]Paolo Giorgini & [2,3]Adi Mulyanto

[1]DIT – University of Trento
via Sommarive 14, Trento Italy
{yudis.asnar,paolo.giorgini}@dit.unitn.it
[2]Informatics Department – Intitut Teknologi Bandung
Ganesha 10,Bandung, Indonesia
adi@informatika.org
[2]P.T. LAPI - DIVUSI
Kyai Gede Utama,Bandung 12, Indonesia
Contact Person:
Yudistira Asnar

Abstract. Modeling and analyzing risk is one of the most critical activities in system engineering. Through this measure, an analyst ensures the security and dependability of a system. In secure and dependable community, Security property is defined as confidentiality, integrity, and availability while dependability with reliability, availability, safety, integrity, and maintainability. These attributes can be achieved by means of controlling the risks that can affect to the system. Risk management is a set of activity that consists of organizational analysis, risk identification, risk assessment, risk evaluation, risk treatment, and risk monitoring.

In this paper, we present several significant works that have been proposed in literature to model and analyze critical information systems (i.e., from infrastructures until organizational structures). Moreover, we also relate them to the risk management process such that can guarantee the achievement of security and dependability properties.

Keywords: *Security, Dependability, Risk Modeling*

1 Introduction

Software systems are more and more part of our life (look how many computers and electronic gadgets are around us), and very often they have a strong influence in our daily life decisions. Considering software systems as integral and active part of an organization introduces the needs of including the software development process as part of the organizational development and not just an incidental project for the organization. This type of software system, usually, is called as critical information system. In literature (Sommerville, 2004), one distinguishes critical system into three classes: safety-critical system, mission-critical system, and business-critical system. To develop a critical system, one can not just provide services that the stakeholders desire, but the system must be able to operate in deviate condition or even when an error occurs.

In this direction, some software engineering methodologies have been proposed. Those methodologies usually based on the requirement analysis or design activity. For requirement analysis, the methodologies, such as Tropos (Liu et al., 2003; Bresciani et al., 2004) and KAOS (van Lamsweerde and Letier, 2000; van Lamsweerde et al., 2003) are found to be useful to analyze the system since early phase of development which consider the relationship between system-to-be and the organizational setting since the early phases of software development. Moreover, Tropos/*i** extension (Liu et al., 2003) introduces the notions of attacker, vulnerability, and countermeasure to help the analyst eliciting robust requirement/design of the system. It is because by analyzing those notions, one can develop a system with considering what stakeholders' desires and anticipating what the "things" that cause a failure. For design activity, some modeling frameworks have been proposed in the literature to model the system and the considerable obstructions that obstructs the system-to-be. Most of those frameworks (e.g., Secure UML (Lodderstedt et al., 2002), UMLSec (Jurjens, 2001), Abuse Case (McDermott and Fox, 1999), and Misuse Case (Sindre and Opdahl, 2005)) extend UML to make it better when modeling a critical system, especially in security aspects (e.g., confidentiality, integrity, and availability). However, most, those frameworks are lack of the analysis process (i.e., qualitatively or quantitatively).

In the safety and reliability community, there are several frameworks that commonly used by reliability engineers to model their systems and assessed the level of safety and reliability. Namely, Fault Tree Analysis (FTA) (Stamatelatos et al., 2002), Failure Modes Effect and Criticality Analysis (DoD), HAZOP (Kletz, 1997), and Reliability Block Diagram (Figiel and Sule, 1990). Those frameworks have been proved to be useful and valid answering the safety and reliability engineering problems. FTA is structured how a failure is developed by representing it as a logic tree, and moreover it can calculated the likelihood of the failure following the probability theory. However, those frameworks, really, concentrate with the system-to-be and often overlook analyzing the organizational where the system will operate.

Essentially, those two approaches show the commonality in analyzing undesired situations from stakeholders' viewpoint and try to mitigate them once the effects are unacceptable. Thus, to this end the notion of risk is, really, appropriate to be used to model and analyze such kind of situations where each undesired situation must be quantified with its likelihood of occurrence and its impact to the system. In this paper, we present the survey about the works that have been done to model and analyze a secure and dependable system. Moreover, we also present several frameworks in risk analysis and management and depict their relations, and hoping that they can complementary in modeling and analyzing a critical information system. Even though, we concentrate mainly explaining the works that have been done for safety critical information system (e.g., air traffic management, healthcare system), but it does not limit the scope of the paper for another type of critical system (e.g., aircraft, nuclear reactor), and surely with minor differences which are domain-depended.

The paper starts from explaining the basic definitions of risk and its properties and several formalisms about uncertainty that have been well defined by mathematician (Section 0). Several modeling and analyzing frameworks are presented either in the area of risk management (Section 3) or critical information system (Section 4). Finally, we present the use of risk analysis realizing a Secure and Dependable (S&D) system using existing frameworks (Section 5) and some conclusion (Section 6).

2 Uncertainty and Risk

It is not certain that everything is uncertain – *Blaise Pascal*

Dealing with the uncertainty is one of the old problem that mankind has tried to deal with. Essentially there are several attempt to model the uncertainty, such as probability theory (Ross, 1997), possibility theory (Dubois and Prade, 2001), fuzzy set (Zadeh, 1978), Dempster-Shaffer theory of evidence (Shafer, 1976), etc. At glace, those theories are similar and can be interchangeable one another, but those assumption is not always true. It is because those theories are developed for different purposes and representing different types of uncertainty. There are several related interpretations of uncertainty:

- **Variability** of such quality of an object. For instance, the computer may crash though we have set it up carefully.
- **Imprecision** (fuzzy/unclear) of the quality of a situation. Such as, the temperature of an engine is hot. Though we have defined that beyond 80°C is hot, we can not simply state that 79°C is normal/not hot.
- **Degree of belief** about a circumstance which is uncertain because of lack of knowledge. Since we do not have complete knowledge about the weather, we can not ensure how the weather is for tomorrow.
- **Likelihood** of a certain condition will be occurred in the future. For instance, the likelihood that the price of stock "X' raises.

Those are several interpretations that may be concluded once we think about an uncertainty. The following passage explains several mathematical theories that model and analyze those concepts.

Probability Theory

This theory are introduced by Kolmogorov (Kolmogoroff, 1950) which define the probability P of an event E (i.e., P(E)), for a given sample space S, must satisfy the Kolmogorov axioms:

$$0 \le P(E) \le 1; \forall E \in S \tag{1}$$

$$P(S) = 1 \tag{2}$$

If E_1 and E_2 are disjoint event, then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$ (3)

The probability value is laid as positive integer value (1). The probability of the event in sample space *S* is certain (equal to 1) (2). The join probability of two events (e.g., E_1 and E_2) which is disjoint is calculated by the addition of the two event probabilities (2).

This theory may represent variability, also called *aleatory probability*, or degree of belief, called *epistemic probability*. Frequentists argue that the probability of an event is relative frequency of an occurrence after repeating a process for large number times. The aleatory probability assumes that an event is resulted by a random phenomenon. The common example of this interpretation is the probability of tossing dice or coin. Thus, we assume that the probability of event *A* is *X* if the relative frequency of event A is equal X for a given huge number *n* trials:

$$\Pr(A) = X \Leftrightarrow \lim_{n \to \infty} \frac{n_a}{n} = X \tag{4}$$

Conversely, Bayesian followers assign the probability to represent the *degree of belief* of the occurrence of a situation. For example, how much is the probability that a suspect really committed a crime. Several mathematicians (e.g., frequentists) oppose the idea of representing degree of belief as a subjective probability as Basyes did. It is because those two uncertainties are founded based on different aspects such that it can fulfill the criteria in (4). Moreover, the probability may conclude the absence of event A ($\sim A$) from$1 - P(A)$. This assumption is too strong for a certain condition. For instance, the probability of *the server will be hacked* is 10%, so that the assumption that *the server will save from hacking* has probability 90% is not really correct. It is because there could be the case that the probability of event *the server will save* is 40%, and the rest (e.g., 50%) can not be decided either save or not, due to the lack of knowledge.

Possibility Theory/Fuzzy Set

This theory is proposed in (Zadeh, 1978) as an alterative mathematical theory to deal with certain types of uncertainty. Essentially, it is introduced as an extension of his theory of fuzzy sets and fuzzy logic. Possibility represents the quality of a measure. For instance, a preposition *the system X is hot* has the possibility value 0.8. It does not imply that the system X has 80% chance to be hot, but it represent in which extent the hotness of system X. For instance, we define that the system X is hot when it is 400°C and normal for 150°C. So that, the possibility 0.8 means that the system X has temperature somewhere around 350°C. The complete axioms are presented in (Zadeh, 1978; Dubois and Prade, 2001), such as:

$$Poss(\phi) = 0 \tag{5}$$

$$Poss(S) = 1 \tag{6}$$

If E1 and E2 are disjoint event, then $Poss(E_1 \cup E_2) = Max(Poss(E_1), Poss(E_2))$ (7)

There is no possibility of the event results on the value which is outside of the sample space *S* (5). It assumes that all the values in sample space S which construct the *Poss* is free from any contradiction (6). Thus, the combination of possibility of two disjoint events is calculated by the maximum value of both possibilities (7).

$$Poss(S) = Poss\left(\bigcup_{x_i \in S} x_i\right) = \max_{x_i \in S}(Poss(x_i)) = 1 \tag{8}$$

In (8), one can conclude inside of the sample space *S* there should be an element with possibility equal to 1 from (7) and (6).

Evidence Theory

It is also called Dempster-Shafer Theory (D-S) (Shafer, 1992). The theory of evidence is used to combine separate pieces of information (evidence) to calculate the probability of an event. Several scholars use this theory to model and analyze the epistemic probability because D-S allows us to represent the lack of knowledge about the domain/sample space. For given possible worlds *S*, D-S allows one to assign the evidence value of one element of *S* (i.e., *m(A)*) or to subset of *S*. Assigning the evidence to a subset happens when the evidence value of a single event is hard to be assessed. For instance, there is enough data/evidence (80%) to decide that the system X will be reliable, and base on past experience, on may decide that the evidence value for the system X to be unreliable is 10%. The remains (10%) are left to be undecided because we do not have adequate information for it.

Suppose assume S is possible words which is $S = \{A, \sim A\}$ thus the powerset of S is $P(S) = \{\phi, \{A\}, \{\sim A\}, \{A, \sim A\}\}$. Thus we may assign the evidence value of the member of $P(S)$ as follow: $m(\phi) = 0, m(A) = 0.8, m(\sim A) = 0.1, m(A, \sim A) = 0.1$. Conversely, this theory has several ways to combine the evidence values depending on the scenario. The complete survey about those rules is presented in (Sentz and Ferson, 2002).

Risk

In ISO (International Standard Organization), **risk** is defined as combination of the probability of an event and its consequences (ISO/IEC Guide 73:2002 definition 3.1.1 "Risk management – Vocabulary – Guidelines for use in standards"). Probability, here, refers to the notion of likelihood that we have explained before. Essentially, the future is uncertain because we do not have complete knowledge to predict it (degree of belief) or it is a random variable where might be measured and analyzed using any statistical methods (variability). Thus, it can be reduced into both notions.

By consequence, an event must deliver any consequences to be considered as a risk. It is irrelevant to consider an event without any consequences. COSO define risk as an uncertain event with negative impact (COSO, 2004). This definition allows us to analyze also opportunity (i.e., uncertain event with positive impact) instead only risk analysis. There are several frameworks that are proposed in the literature to model and analyze risk, either qualitatively (DoD, 1980) or quantitatively (Bedford and Cooke, 2001). The following sections will present some of those frameworks.

3 Risk Management

There are several confusions between among the concepts of risk analysis, risk assessment, risk evaluation, and risk management. In this paper, we establish the definition for those concepts following existing standards. **Risk management process** is a systematic application of management policies, procedures and practices to the tasks of establishing the context, identifying, analyzing, evaluating, treating, monitoring, and communicating risk (AS/NZS 4360:1999)[1] or a continuous process for systematically identifying, analyzing, treating, and monitoring risk throughout the life cycle of a product or service *(*ISO/IEC 16085). This process is really central to define the response of an enterprise towards the existing risk. In ISO/IEC Guide 73, **Risk assessment** is defined as an overall process of risk analysis and risk evaluation. **Risk analysis**, itself, contains of several steps:

- **risk identification** identifying uncertain events that may cause organization's exposure;

[1] Australian Standard / New Zealand Standard

- **risk description** detailing and structuring identified events;
- **risk estimation** defining qualitatively or quantitatively the probability of occurrence and the possible consequence of identified events.

Risk evaluation is used to make decisions, such as whether an event is acceptable or need to be treated. Based on this activity, an enterprise has to define which countermeasures must be employed to mitigate the unacceptable risks/events. However, one may consider the possibility of loosing an opportunity once risks are mitigated.

CORAS (den Braber et al., 2003) and Enterprise Risk Management-COSO (COSO, 2004) are some of the frameworks that are proposed in the literature as risk management framework. CORAS are developed mainly for secure critical information system. ERM-COSO is developed to manage the risk in the enterprise mainly in the aspects that are related to financial.

3.1 Enterprise Risk Management

This framework has been developed by PricewaterhouseCoopers and Committee of Sponsoring Organizations of the Treadway Commission (COSO, 2004) which is defined as a process that:

- is effected by every people at every layer of the enterprise;
- is applied in strategy setting and across the enterprise;
- is designed to identify potential events that may affect the enterprise;
- manages the risk to be within the enterprise risk appetite[2];
- provides reasonable assurance regarding the achievement of the enterprise objectives.

The framework has three dimensions: achievement of objectives, components of ERM, and entities. Entity is defined as an organization that is purposed to provide some values to its stakeholders (e.g., business enterprise to run the business of shareholders). The framework categorizes achievement of objectives as follows:

- Strategic -high-level goals, aligned with and supporting the mission of entity;
- Operation -effective and efficient use of the resources of entity;
- Reporting -reliability of reporting;
- Compliance -entity compliance to the existing laws, regulations, and standards.

An objective can be categorized in more than one category. The categorization aims to address different entity interests and to identify the expectation of an entity to each category. Since objectives related to *reporting* and *compliance* are within the entity control (i.e., its achievement depends on how well the related activities are performed by entity), ERM provides a reasonable assurance for the achievement of these objectives. Differently, the achievement of objectives in *strategic* and *operation* category is related to external events which are beyond the entity control. ERM can only provide, in this case, reasonable assurance in which the entity is moving toward the achievement of objectives.

[2] the amount of risk that an entity is willing to accept in pursuit of its mission/vision

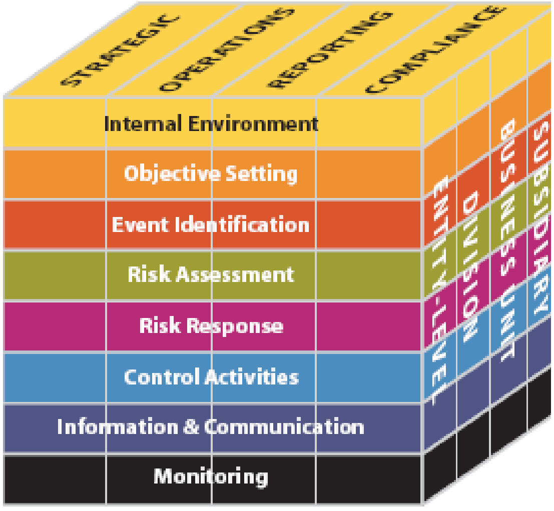

Figure 1 COSO Framework[3]

ERM is composed of eight interrelated components that can be viewed as steps to manage risk. The components are:

- *Internal Environment*, it is the foundation of all ERM components. It consists of how risks are viewed and addressed by people in entity. Moreover, it includes risk management philosophy, risk tolerance, integrity and ethical values in entity;
- *Objective Setting,* it is a setting of high-level goals which are aligned with, and supports entity visions/missions. Through this setting, the management can consider any alternative strategies to achieve strategic objectives considering the associated risks;
- *Event Identification,* an event is identified from internal or external sources of entity. It could affect the implementation of the strategy and the achievement of objectives. Events may have positive impact (called opportunity), negative impacts (called risk), or both. ERM considers the occurrence of an event as non-isolated occurrence because an event can trigger other events or they can occur concurrently.
- *Risk Assessment,* it assesses an event from two perspectives: likelihood and impact. It allows an entity to consider which event might have significant influence to the achievement of the objectives.
- *Risk Response,* it defines how entity will response to the risk. It includes risk avoidance, risk reduction, risk sharing, and risk acceptance. The management considers several factors in choosing the proper risk responses, such as:
  - Impact of a risk response on reducing risk likelihood or impact;
  - Cost and benefit of potential risk response;
  - Possible opportunities lose, once we apply risk response.
- *Control Activities,* policies and procedures are defined to ensure that risk responses are carried out once risks occur. In (COSO, 2004), COSO has defined types and several common control activities that are usually applied in risk management.
- *Information and Communication,* every enterprise identifies and captures relevant information to managed entity. This information needs to communicate to the entire entity personnel, in a certain form and certain time frame, to help them in performing their responsibilities and roles in ERM.

[3] COSO (2004). Enterprise Risk Management - Integrated Framework, Committee of Sponsoring Organizations of the Treadway Commission.

- *Monitoring,* it is a process to assess the current presence and function of the ERM components. An entity in ERM might change over time; the entity objective can change. This fact could cause a risk, and its risk response, becoming irrelevant.

ERM covers, nearly, all steps of risk management process without any precise guidance of each step. We consider this work as a global framework for: informal guidance (Holton, 2004), basic taxonomy (Carr et al., 1993), and existing models/techniques (DoD, 1980; Kletz, ; Stamatelatos et al.). Moreover, in (Marchetti, 2005) the author motivates that by applying ERM an enterprise my also achieve the compliance with section 404 of Sarbanes-Oxley Act.

3.2 CORAS

CORAS is a research and technological development project under Information Society Technologies (IST) Program that is aiming at developing a framework for risk analysis of security critical systems. The CORAS risk management consists of the following steps (Coras, 2005):

- *Context Establishment*, by selecting usage scenarios of the system;
- *Ris*k *Identification,* tries to identify the threats based of scenarios and the vulnerabilities of these assets;
- *Ris*k *Analysis*, assigns the values (impact and likelihood of occurrence) to identified threats;
- *Ris*k *Evaluation*, identifies the risk level of each threat;
- *Ris*k *Treatment*, addresses the treatment of considered threats.

Moreover, CORAS adopts UML as a modeling language and extends for modeling a security critical system. For each step, CORAS may adopt several existing analysis tools that has commonly used (e.g., FMECA (DoD, 1980), FTA (Stamatelatos et al., 2002), HAZOP (Kletz, 1997)). Those analysis tools will be briefly explained in the following section and the guidelines for choosing and integrating these tools into CORAS steps has been presented in (Coras, 2005), and it will be updated further during the progress of the CORAS project. The methodology has already been tested in security systems (especially E-Commerce and Telemedicine).

Generally, all steps in CORAS have already been covered by ERM. However, risk identification phase in CORAS is unique, which is done on the basis of the execution scenarios and not based on the vision and mission of an enterprise.

4 Critical Information System

Critical Information System (CIS) is a type of information system (IS) where failure may cause severe human or economic consequences (Sommerville, 2004). Essentially, there are three types of CIS:

- Safety-critical system – failure may result in the loss of life or damage to the environment direct or indirectly, such as the nuclear plant explosion.
- Mission-critical system – failure may cause to the failure of some activities that are means to achieve the goals, for instance the aircraft navigation system.
- Business-critical system – failure may result to the high economic losses, as the electronic banking system.

Those information systems must be analyzed, designed, developed, tested, and maintained carefully such that the failure might be avoided. Essentially, this type of system is required to secure and dependable all the time of their operation. Actually, security and dependability are two major qualities that must be provided by such kind of information system.

Security computing (Pfleeger and Pfleeger, 2006) is defined as an aggregate property which consists of Confidentiality, Integrity, and Availability. By *confidentiality*, it ensures that the system might be accessed only by authorized users. *Integrity* means preventing the system for improper or unauthorized change. *Availability* refers to the ability to use the system when it is needed. To enforce such properties, the designers must employ additional mechanisms to protect the system from hostile behavior.

In the security engineering, the designer must model the system and verify whether the model satisfies security properties that are required for such information system. For instance, healthcare information must protect patient's data from improper modification and forgery. Moreover, not all the doctors are able to modify the patient records. Doctors are able to modify the patient records that are assigned as their patients, otherwise they does not such kind of authorization to do so. For this end, the system must have the login system to prevent the modification from unauthorized users. Moreover, designers must identify which kind of threat or vulnerability that might exist and effect to the system. For instance, the designer may identify a threat such as a patient might want to modify his health record becoming the better one, such that he can be employed in a dangerous sector.

There are several modeling frameworks have been introduced to model and analyze the system, such that its security is guaranteed. In (Dardenne et al., 1993), the author proposes KAOS, a goal-oriented requirements engineering methodology which models not only *what* and *how* aspect of requirements but also *why*, *who*, and *when*. Moreover, KAOS introduces also the concept of *obstacle* (van Lamsweerde and Letier, 2000) and *anti-goal* (van Lamsweerde et al., 2003) which can be seen as boundaries in requirement analysis. An obstacle is defined as an undesirable behavior to strategic interests of stakeholders, and an anti-goal defines a goal that belongs to an attacker that obstructs the fulfillment of stakeholders' goals. In other word, obstacles can be seen as unintentional-risk, since risk is an undesirable behavior, and anti-goals are threats or intentional risks. These features make KAOS suitable for analyzing requirements of secure and dependable system.

The authors (Mayer et al., 2005) extend the *i** conceptual framework (Yu, 1995) to analyze risk and security issues during the development process of IT systems, requirement analysis in particular. The framework models the business assets (i.e., goals) of an organization and assets of its IT system (i.e., architecture, design decisions). Countermeasures to mitigate risks are then selected in such a way that the risks do not affect these assets. The proposal (Liu et al., 2003) proposes a methodological framework for security requirements analysis based on *i**. They use the NFR framework (Chung et al., 2000) to support the formal analysis of threats, vulnerabilities, and countermeasures. There is also approaches like *attack tree* or *threat tree* (Schneier, 1999; Helmer et al.) which are similar to the FTA. Others proposals like UMLSec (Jurjens, 2001), SecureUML (Lodderstedt et al., 2002), Abuse Case (McDermott and Fox, 1999), and Misuse Case (Sindre and Opdahl, 2005) are funded on UML as modeling language which has commonly used in the computer society. Unfortunately, those modeling framework overlook to model either the likelihood of a threat or the consequence of a threat or both. Their main role is to identify the vulnerabilities and the threats in the system and structure them. By means of this result, a designer may conclude weather necessary or not a countermeasure to be introduced.

In dependability community, **dependability** (Avizienis et al., 2004) is an aggregate concept that encompasses attributes: availability, reliability, safety, integrity, and maintainability. This concept somehow is overlap with the one from security. Security does not consider the importance of the safety aspect, while dependability does not put any concerns about protecting the confidentiality aspect of the system. *Reliability* is defined as continuity to provide correct service. *Safety* is the absence of catastrophic consequences on the human life or environment. *Maintainability* means the ability to be modified or repaired.

There are several old works, such as FMECA, FTA, and HAZOP that has commonly, can be used to assist the designer modeling and analyzing the system. FMECA and FTA have been quantified the likelihood of a failure. Moreover, FMECA analysis specifies what the impacts are once the failure occurs. Moreover, there is a framework proposal (Feather, 2001), called Defect Detection and Prevention-DDP, which is developed and applied in JPL and NASA. The DDP model also allows us to work together with other quantitative tools (e.g., FMECA, FTA) to model and assess risk/failure. In (Feather et al., 2002), they demonstrate how to integrate FMECA and FTA into DPP model. DDP consists of three layers model (Objectives, Risk, and Mitigation). *Objectives* are the things that the system has to achieve. *Risks* are all the kinds of things that, once occur, lead to the failure of objectives achievement. Finally, *mitigations* are action that can be applied to reduce the risks. Despite these, there are two relations that relate risk to objective, and mitigation to risk, namely impact and effect. Impact is defined as the quantitative representation of objective lost once the risk occurs. Effect is defined as quantitative representation of risk reduction if the mitigation is applied. In this model, each objective has a weight to represent its importance. A risk has a likelihood of occurrence and mitigation has a cost for accomplishment (namely resource consumption). Moreover, DDP model specifies how to compute the level of objectives achievement and the cost of mitigations3 from a set of taken mitigations. This calculation allows us to evaluate the impact of the taken countermeasures and thus supports the decision making process. However, DDP depicts objective (also risk and mitigation) as a solitary object (i.e., there is no relation among objects in the same layer). This feature could not model the situation (e.g., domino effect (Reniers et al., 2004)) in which the occurrence of a risk can increase/decrease the likelihood of other risks. Thus, in (Asnar et al., 2006) we propose a similar framework that overcome this limitation and the framework can be operated in the early phase of system development (i.e., requirement analysis), so that the designer is able to anticipate the risky requirements.

In the area of risk modeling, uncertain events (e.g., threats and failures) are quantified with two attributes: likelihood and severity. Probabilistic Risk Analysis (PRA) (Bedford and Cooke, 2001) is widely used for assessing risk quantitatively, while FMECA quantifies risk into qualitative values: frequent, reasonable probable, occasional, remote, and extremely unlikely. Essentially, events are prioritized using the notion of *expectancy loss* which is a multiplication between the likelihood of events and its severity. This priority represents the criticality of an event. When resources are limited, an analyst can decided to adopt countermeasures for mitigating events on the basis of their priority. However, identification of probabilities is not necessarily precise, and typically it strongly depends on expert judgments. Approaches like Multi-Attribute Risk Assessment (Shawn and Paul, 2001) can improve the risk analysis process by considering multi-attributes. Many factors like reliable, available, safety and confidentiality can result critical for a system and each of them has its own risk value. This introduces the need for the analyst to find the right trade-off among these factors. For instance, an Air Traffic Management system is required to be always available and safe. Certain conditions (e.g., radar noise) can affect the normal behavior of the system and consequently its safety. In many cases, the best solution is to restart the system. This, however, reduces the availability of the system. In (Butler, 2002), the author presents how to choose cost-effective countermeasures to deal with existing security threats by using multi-attribute risk assessment.

5 From Risk Analysis to Secure and Dependable System

Common people often find a difficulty distinguishing among the attributes of security and dependability. Essentially, those attributes are alike but they are not exactly the same. The same confusion is also created in distinguishing failure, hazard, or threat with the notion of risk. Therefore, in this section we try to clarify the distinction of those concepts and how to use the notion of risk to model and guarantee the achievement of security and dependability properties.

The most common confusion is created between availability and reliability. Essentially, those properties are alike in certain aspect. Such as, they represent the quality of a correct service of the system. Availability commonly represents by the notion of *mean-time-to-failure* (MTTF), which represents the average time before the system fails, and *mean-time-to-repair* (MTTR), which is the mean time that is needed to repair the system from a failure. A high reliable system may cause the high degree of availability because high reliability will increase MTTF of the system. However, to achieve the high availability one can achieve by having an easy maintenance system, because it will result a lower MTTR once the failure occurs. In certain case, it is so costly for having a high reliable system, because it might be achieved by employing another duplicate system as a redundant in case the main system fails and it is not the case when the designer tries to have a system with low MTTR.

Some engineers may argue that all properties can be reduced into reliability and availability property. In the case of safety, typically, they assume that by having the system is well operated than the catastrophic consequences can be avoided. This argument makes sense in many cases but it is not always. There could be the case that engineers should choose between maintaining the availability or the safety. For instance, in the case of nuclear power plant it could be the case that the reactor is overheated. If the engineers decide to maintain the availability, then they will set the reactor operating in degraded mode, which may lead to catastrophic consequences. In case the engineers decide to maintain the safety, then the easy way to achieve it is by shutting down the plant. The similar motivation might be applied for other properties of security and dependability.

The risk notion is adopted to ensure until which extent the system is secure or dependable. Having the system 100% secure and dependable is an utopia. It is because the system will be so costly to be implemented. Moreover, we are not able to implement the stakeholders' requirements because they can create vulnerability and lead to the system to be compromised. For instance, we do not want there is an attack to our finance system. The most convincing way to have it is by not allowing external access to the finance system. This measure, surely, contradicts to the requirement of allowing branch office to access their ledger in the finance system. Therefore, the designer must assess the likelihood of a failure or an attack and predict the possible losses that are introduced by them. Later, the designer can perform trade-off analysis to decide whether employing additional mechanisms is cost effective or not (i.e., the expectancy loss is equal-higher that the additional investment).

Suppose we have a service that is required to be 99.99% to be available. There are several possibilities to implement this requirement: 1) having the service with 4 replicas, 2) having the service with 2 replicas and assigning dedicated personnel to repair the system. By calculating the possible loss in case the system is not available, one may compare which option is the most cost-effective.

In the case of security engineers, the designer typically faces enormous number of threat and vulnerability. It makes impossible to address all of them and adding a new mechanism as a response. The notion of risk can be adopted to prioritize among them and allocate the resources (e.g., investment, cost, etc.) following this prioritization, and could be we ignore the insignificant threat or vulnerability, in terms of their likelihood is unlikely or their consequences is not severe.

6 Conclusion

Though, this paper is not presenting new contribution in the area of risk analysis or security and dependability engineering, but we try to address in which aspects those fields are intersect. We explain several types of uncertainty and the basic principle of risk. Risk may represent an uncertainty because of variability or degree-of-belief. The variability appears when we analyze a hardware system where each behavior is predefined, such that we can use statistical methods to calculate the uncertainty. The degree-of-belief is commonly used when the experience data is not adequate such that it needs the judgment of the experts.

Moreover, we clarify the distinction among security and dependability properties and relate them with the notion of risk. We have presented the state-of-the-art that has been done in the scientific or practice world. We also motivate how to use the risk analysis complementary with security and dependability framework such that the designer can prioritize the available resource to response the malicious situations according to their prioritization.

Acknowledgement

This work has been partially funded by EU Commission, through the SENSORIA and SERENITY projects, by the FIRB program of MIUR under the ASTRO project, and also by the Provincial Autonomous of Trento, through the MOSTRO project.

References

Asnar, Y., P. Giorgini & J. Mylopoulos (2006). Risk Modelling and Reasoning in Goal Models, DIT - University of Trento.

Avizienis, A., J.-C. Laprie, B. Randell & C. E. Landwehr *Basic Concepts and Taxonomy of Dependable and Secure Computing*. IEEE Transactions on Dependable and Secure Computing **1**(1): 11-33.(2004)

Bedford, T. & R. Cooke *Probabilistic Risk Analysis: Foundations and Methods*, Cambridge University Press (2001)

Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia & J. Mylopoulos *Tropos: An Agent-Oriented Software Development Methodology*. Journal of Autonomous Agents and Multi-Agent Systems **8**(3): 203--236.(2004)

Butler, S. A. *Security Attribute Evaluation Method: a Cost-Benefit Approach*, in *Proceedings of the International Conference on Software Engineering (ICSE'02)*, New York, NY, USA, ACM Pres (2002)

Carr, M. J., S. L. Konda, I. Monarch, F. C. Ulrich & C. F. Walker (1993). Taxonomy-Based Risk Identification, Software Engineering Institute, Carnegie Mellon University.

Chung, L. K., B. A. Nixon, E. Yu & J. Mylopoulos *Non-Functional Requirements in Software Engineering*, Kluwer Publishing (2000)

Coras (2005). CORAS: A Platform for Risk Analysis of Security Critical System, http://www.nr.no/coras/.

COSO (2004). Enterprise Risk Management - Integrated Framework, Committee of Sponsoring Organizations of the Treadway Commission.

Dardenne, A., A. van Lamsweerde & S. Fickas *Goal-Directed Requirements Acquisition*. Science of Computer Programming **20**: 3-50.(1993)

den Braber, F., T. Dimitrakos, B. o. r. A. Gran, M. S. Lund, l. K. St\o & J. O. y. Aagedal *The CORAS Methodology: Model-Based risk assessment using UML and UP*.in UML and the Unified Process, Idea Group Publishing**:** 332--357. (2003)

DoD *Maintainability Prediction (MIL-HDBK-472)*, U.S. Department of Defense (1966)

DoD *Military Standard, Procedures for Performing a Failure Mode, Effects, and Critical Analysis*, U.S. Department of Defense (1980)

Dubois, D. & H. Prade *Possibility Theory, Probability Theory and Multiple-Valued Logics: A Clarification*. Annals of Mathematics and Artificial Intelligence **32**(1-4): 35--66.(2001)

Feather, M. S. *Risk Reduction Using DDP (Defect Detection and Prevention): Software Support and Software Applications*, in *Proceedings of the 6th IEEE International Symposium on Requirements Engineering* (2001)

Feather, M. S., S. L. Cornford, J. Dunphy & K. Hicks *A Quantitative Risk Model for Early Lifecycle Decision Making*, in *Proceedings of the Conference on Integrated Design and Process Technology* (2002)

Figiel, K. D. & D. R. Sule *A generalized reliability block diagram (RBD) simulation*, in *Proceedings of the 22nd Conference on Winter Simulation (WSC'90)*, Piscataway, NJ, USA, IEEE Press (1990)

Helmer, G., J. Wong, M. Slagell, V. Honavar, L. Miller & R. Lutz *A Software Fault Tree Approach to Requirements Analysis of an Intrusion Detection System*. Requirements Engineering Journal **7**(4): 207--220.(2002)

Holton, G. A. *Defining Risk*. Financial Analyst Journal **60**(6): 1925.(2004)

Jurjens, J. *Towards Secure Systems Development with UMLsec*, in *Proceedings of the 4th International Conference on Fundamental Approaches to Software Engineering*, Springer-Verlag (2001)

Kletz, T. A. *HAZOP - Past and Future*. Reliability Engineering and System Safety **55**(3): 263-266.(1997)

Kolmogoroff, A. *Foundations of the theory of probability.-Translated by N. Morrison*.(1950)

Liu, L., E. S. K. Yu & J. Mylopoulos *Security and Privacy Requirements Analysis within a Social Setting*, in *Proceedings of the 11th IEEE International Requirements Engineering Conference* (2003)

Lodderstedt, T., D. Basin & J. Doser *SecureUML: A UML-Based Modeling Language for Model-Driven Security*, in *Proceedings of the 5th International Conference on the Unified Modeling Language -- the Language and its Applications*, Springer-Verlag (2002)

Marchetti, A. M. *Beyond Sarbanes-Oxley Compliance: Effective Enterprise Risk Management*, John Wiley & Sons (2005)

Mayer, N., A. Rifaut & E. Dubois *Towards a Risk-Based Security Requirements Engineering Framework*, in *Proceedings of the 11th International Workshop on Requirements Engineering: Foundation for Software Quality* (2005)

McDermott, J. & C. Fox *Using Abuse Case Models for Security Requirements Analysis*, in *Proceedings of 15th Annual Computer Security Applications Conference*, Phoenix, AZ, USA (1999)

Pfleeger, C. P. & S. L. Pfleeger *Security in Computing*, Prentice-Hall (2006)

Reniers, G., W.Dullaert & K. Soudan (2004). A Domino Effect Evaluation Model, University of Antwerp, Faculty of Applied Economics.

Ross, S. M. *Introduction to Probability Models*, Academic Press (1997)

Schneier, B. *Attack Trees: Modeling Security Threats*. Dr. Dobb's Journal **12**(24): 21--29.(1999)

Sentz, K. & S. Ferson *Combination of Evidence in Dempster-Shafer Theory*.(2002)

Shafer, G. *A Mathematical Theory of Evidence*. Princeton, NJ, Princeton University Press (1976)

Shafer, G. *The Dempster-Shafer theory*.in Encyclopedia of Artificial Intelligence. S. C. Shapiro, Willey**:** 330-331. (1992)

Shawn, B. & F. Paul (2001). Multi-Attribute Risk Assessment, Carnegie Mellon University.

Sindre, G. & A. L. Opdahl *Eliciting Security Requirements With Misuse Cases*. Requirements Engineering Journal **10**(1): 34--44.(2005)

Sommerville, I. *Software Engineering*, Addison Wesley (2004)

Stamatelatos, M., W. Vesely, J. Dugan, J. Fragola, J. Minarick & J. Railsback *Fault Tree Handbook with Aerospace Applications*, NASA (2002)

van Lamsweerde, A., S. Brohez, R. D. Landtsheer & D. Janssens *From System Goals to Intruder Anti-Goals: Attack Generation and Resolution for Security Requirements Engineering*, in *Proceeding. of RHAS'03* (2003)

van Lamsweerde, A. & E. Letier *Handling Obstacles in Goal-Oriented Requirements Engineering*. IEEE Transactions on Software Engineering **26**(10): 978--1005.(2000)

Yu, E. (1995). Modelling Strategic Relationships for Process Engineering, University of Toronto, Department of Computer Science.

Zadeh, L. A. *Fuzzy sets as a basis for a theory of possibility*. Fuzzy Sets and Systems **1**: 3-28.(1978)