# Multi-Dimensional Uncertainty Analysis in Secure and Dependable Domain

Yudistira Asnar
Department of Information Science and Engineering
University of Trento, Italy
Email: yudis.asnar@disi.unitn.it

Paolo Giorgini
Department of Information Science and Engineering
University of Trento, Italy
Email: paolo.giorgini@disi.unitn.it

***Abstract*—Most of the critical aspects for secure and dependable systems, such as safety, integrity, availability, are related to uncertainty. Literature proposes many approaches to deal with uncertainty, mainly in the area of risk management and safety&reliability engineering. However, what is still missing is a clear understanding of the nature of uncertainty that very often has produced mistreatments in the design. In this paper, we propose a conceptual model for uncertainty that can be used to deal with systems' qualities such as security and dependability. Particularly, we will consider the relation between uncertainty-risk and how risk affects quality attributes of the system. We use a case study in Air Traffic Management to illustrate our approach.**

## I. Introduction

Software systems are assuming more and more a critical role in our daily life and this introduces the need for software developers to deal more deeply with problems related to security and dependability (S&D). Several approaches have been proposed in literature to develop secure and dependable systems [1]–[5] and almost all of them focus on how preserve their properties in all possible situations. Avizienis et al. [1] proposed a taxonomy to clarify the basic concepts of security and dependability that can be considered complementary to what has been proposed in [2], [6]. Both, however, consider risk as an uncertain event that produces a negative impact into the system [7] and that may obstruct its normal behaviour.

Techniques based on ontologies and taxonomies, such as [4] and [3], have been proposed in literature to identify risk, but unfortunately they have not produced significant results for designing of secure and dependable systems. The main problem here is about how to assess *likelihood* and *severity* of risk and what kind of treatment has to be introduced to make risk acceptable for the overall design. In the dependability research community, Fault Tree Model (FTA) [5] is used to structure how a failure occurs and then to assess the probability of the failure occurrence. FTA uses *probability theory* to compute the probability of failure. Similarly, in the security area the attack graph [8] is proposed to model how an attack is conducted and, following Bayes Network rules, what is its probability. All these approaches analyse and assess uncertain events (i.e., risks, attacks, faults, errors) in terms of the likelihood. The common idea is to ensure the system is "good enough" in preserving its security and dependability under uncertainty conditions. However, there is not a clear understanding about which uncertain events one should consider. We can consider, for example, different events like

- a controller fails after 10 days of operation and this happens in 90% of cases;
- 30% of software code introduces vulnerabilities;
- a controller is hot.

All these events can be considered as risks for a system, but each of them has a different nature of uncertainty, and consequently introduces specific needs in terms of treatment. The first one refers to the variability of the controller [9], the second to the lack of knowledge [10] in detecting the vulnerability of code, and the third one to an imprecise definition of "being hot".

In this paper, we propose a conceptual framework to model and analyse risk within the context of security and dependability. We clarify about different natures of uncertainty (i.e., aleatory, epistemic, and fuzziness/impreciseness) and how they influence properties of security and dependability. The models, we propose, can be used by an analyst to identify risks and evaluate their impact over the system according to their nature. Since security and dependability are aggregation of different quality attributes (e.g., availability, confidentiality, integrity, etc.), in our approach the risk assessment must be conducted from different dimensions.

The remaining of the paper is structured as follows. Initially, we present our understanding of main security and dependability concepts. We identify several types of uncertainties (Section III and propose a generic framework for risk assessment that is able to accommodate all types of uncertainty (Section IV). Finally, we discuss our proposal and draw some remarks (Section V).

## II. Security and Dependability: Basic Concepts

In this section, we clarify our understanding about security and dependability as quality attributes. Starting from two US-DoD standards: Orange Book [11] and Failure-Modes and Criticality Analysis (FMECA) [12] that are considered referential works for S&D engineering community.

### A. Quality Attributes

In Fig. 1, we propose a taxonomy for security and dependability as quality attributes. It is mainly based on the work

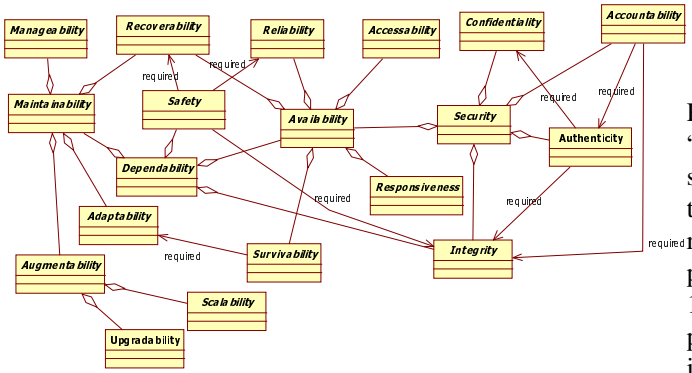


Fig. 1. Security and Dependability Properties

of Avizienis et al. [1] and extended with other approaches proposed in [2], [6], [13], [14].

Information security [15] is refined in terms of confidentiality, integrity, availability, and according to the ISO/IEC 1335 [16] also authenticity and accountability. Dependability is defined as an aggregation of availability, reliability, integrity, safety, and maintainability [1]. More precisely:

- *Availability* is the readiness of a system (or a component) to perform its functionalities. It can be:
  - High *reliability* - continuity of providing a correct service for a specific period;
  - Short *recovery* - repairing the system from a failure and restoring to the "good" condition before the failure occurs;
  - High *survivability* - delivering the functionalities under attacks/failures;
  - High *accessibility* - accessible by users in delivering the functionalities;
  - High *responsiveness* - users receive the response in timely manner.

  *Responsiveness* and *accessibility* attributes refer more to the users' perception than the response of the system [17].
- *Confidentiality* - information can only be accessed by authorized users;
- *Integrity* - the absence of unauthorized/improper system alterations. It also subsumes the absence of data modification without authorization;
- *Authenticity* - ensuring the validity of a subject identity;
- *Accountability* - ability of a system of tracing back any actions;
- *Maintainability* - possibility to change the system after it has been deployed. As in [1] and [6], maintainability is refined into more fine-grained qualities:
  - *Manageability* represents the easiness of managing the system;
  - *Recoverability*, as defined for availability;
  - *Augmentability* is the easiness to upgrade the system as consequence of obsoleteness or scalability problems;
  - *Adaptability* is the easiness to adjust the system according to the environment situation
- *Safety* depicts the absence of catastrophic consequence to users/environment.

Relations between these qualities are represented in Fig. 1. The "required" relation indicates that to realize a property (e.g., safety) the system requires another property (e.g., reliability) to be guaranteed also. We highlight only minimal required-relations; there could be specific situations where the safety property requires more properties than those depicted in Fig. 1. Composition relations (diamond arrows) represents that a property is a composition of several other properties. For instance, to realise a safe system, the design the system should guarantee highly reliability, short recovery time, and integrity. For an accountable system, it is required to have authentication mechanisms and ability to preserve integrity of auditing trails.

S&D attributes are related also to other quality attributes of the system (e.g., usability and performance). For example, security usually makes the system less usable. Generally, relations among attributes are represented as causal-effect [18] or contribution relations [19]. However, this could be not enough. Indeed, the relation between safety and limited-usability is a correlation relation, and not contribution nor causal-effect one. The "cause" of having limited-usability depends on the specific security mechanism that has been chosen, and not on the state of being secure. In other word, there could be a mechanism which does not limit the system usability. For example, in a system that has to protect against Denial-of-Service (DoS) attacks (security), the use of a firewall that limits the users' access will surely secure the system and also limit the usability. Conversely, having Intrusion Detection System (IDS) can mitigate DoS attacks without limiting users' access.

### B. *"The Attributes" of S&D systems*

As emerging from literature, reliability, confidentiality, integrity, and recoverability are the "most" important S&D properties[1]. In the following, we show by examples how other important S&D quality attributes can be reduced to these four properties.

**Example 1.** To maintain the availability of an ATM system, analysts should ensure that the system is highly reliable and requires a short period of time for recovering from failures.

Though we do not assess the accessibility and responsiveness, we may still argue about the availability of the system. However, the availability is often measured by the percentage of operated-time over a period-time. In this way, a *reliable* system may result in a low frequency of downtime, and a *recoverable* system allows us to have a short-period for each downtime.

**Example 2.** To ensure the safety of an aircraft, controllers require the sustained correct assistance (i.e., reliable and recoverable) of an ATM system, and only authorized personnels are allowed to access the system (i.e., integrity)

[1]In this work, we mainly focus on the first three quality attributes

Notice, other aspects of an available system (i.e., reachability, responsibility, or survivability) are not sufficient to guarantee the safety operation in ATM. For instance, a high survivability system means the system keeps running though an attack/an error occurs. Actually, it is dangerous for the safety because the system usually runs in a degraded mode where the result is not 100% correct, and can give mislead information to the controllers.

**Example 3.** For authentication purposes, each controller has different ratings that indicate their experiences. A Novice controller should not be allowed to manage highly-dense airspace (e.g., approach/TRACON sectors). For this end, an ATM system should authenticate users before they use the system.

This setting requires the system to be provided with an access control (i.e., authentication mechanisms and authorization managements). In [13], the authors mentioned that there are three means to authenticate users: 1) by what users know - password, 2) by what users have - card, and 3) by what users are - biometric. Besides ensuring the *reliability* of the access control, to guarantee the authentication the system should preserve the *confidentiality* of authentication means (e.g., passwords, cards) and preserve their *integrity* against illegal modification. When the integrity of authentication means is compromised, the authenticity of users is doubtful though the access control still runs correctly.

**Example 4.** Controllers' actions should be accountable, especially, during an aircraft action. An ATM system always produces audit-trails for every action of controllers and conversations between controllers and pilots. These trails are useful to perform an investigation for any incidents/accidents.

To guarantee this property, the system should provide: 1) an *authentication* mechanism to ensure the authenticity of controllers and 2) maintain the *integrity* of the audit trails from illegal modifications or fabrications [13].

Finally, these reductions, into four main qualities, are useful to reduce the number of assessments required by a system. However, it may overlook on capturing particular attacks or incidents (e.g., DoS attack). In that attack, the system is still reliable and consequently does not require any recovery but it is not available to users. For maintainability issues, we believe it is hardly possible to reduce them into reliability, confidentiality, integrity, and recoverability properties. Ideally, the maintainability should be inherent as main principles in developing the system that should be keep in mind by the developers (i.e., analysts, designers).

## III. The Nature of Uncertainty

In standards like [7] and [20], risk is defined as the combination of the probability (uncertainty) of an event and its (negative) consequences. In other words, risk has two properties: uncertainty and severity. An event, which is certain or has no negative impacts to the system, should not be considered as risk, but more as problem.

*... Uncertainties appear everywhere. ... When using a mathematical model careful attention must be given to uncertainties in the model.* - R. P. Feymann [2]

Besides assessing events' impacts, it is essential to know the uncertainty associated to an event and what is its nature. Several works have been proposed about the nature of uncertainty [21]–[23] and [24]. In [23], the authors argue that essentially uncertainty is divided into two classes: aleatory and epistemic uncertainty. The former is induced by randomness, and the latter is due to incomplete knowledge. In another paper [22], Smithson argues that uncertainty is introduced because of vagueness. Uncertainty is resulted from ambiguity, vagueness (or impreciseness), and probability. In this paper, we consider uncertainties into three classes: aleatory, epistemic, and imprecise.

Uncertainty is always present though one perceives having a complete knowledge. For instance, we know all about a dice including possible outcomes, but still we cannot decide for certain the outcome of rolling a dice. In this example, we can categorise the event of rolling dice is exposed to an aleatory uncertainty induced from the nature of **aleatory** (i.e., variability and randomness).

**Example 5.** Radar engineers have, almost, full knowledge about how a radar works, but they use another radar as a backup because the main radar might fail.

This class of uncertainty is heavily studied in the dependability community. Typically, the uncertainty arises due to spatial variation (e.g., the development site is different with the runtime site), temporal fluctuation, and development variability. Since this uncertainty may result in failures, analysts should investigate all possible uncertain events in this class. The Probabilistic Risk Analysis (PRA) [9] or fault tree analysis (FTA) [5] are useful in assessing the level of uncertainty. However, they cannot reduce the uncertainty. To deal with such uncertainties, redundancy techniques [25] (e.g., primary-secondary, main-backup, multi-version programming) are used to deal with these events.

**Epistemic** uncertainty is caused by incomplete knowledge.

**Example 6.** Aircraft hand-off procedures to adjacent sectors might fail though the ATM system operates normally. It is because analysts do not have complete knowledge about all possible actions that a controller will do for handing-off an aircraft.

Most of security problems are arisen because of the incompleteness. Security analysts are hardly possible aware of all possible threats that attackers might launch to the system. Essentially, there are two subtypes of epistemic uncertainty: 1) visible incompleteness where *we are aware about what we do not know*, and 2) blind incompleteness where *we are not aware about what we don not know*. The example 6 is

[2] in Appendix F - Report of the Presidential Commission on the Space Shuttle Challenger Accident (In compliance with Executive Order 12546 of February 3, 1986), NASA, 1986

one example of visible incompleteness. Essentially, it can be reduced by empirical efforts, such as: interviews, collect more data, expert judgements, etc.

**Example 7.** In smart-card technology, attackers can extract secret keys and consequently compromise the integrity of smart-card by analysing the power consumption of a smart-card [26].

In this setting, analysts never had got an idea how such possibilities existed. Essentially, there is no such mechanism to reduce this subtype of uncertainty. Having a system with high augmentability is beneficial for this uncertainty because it should allow us to easily upgrade the system as soon as vulnerability is discovered. Both types of uncertainty are also introducing risks to the system, but they require different treatments to reduce/to deal with the nature of their uncertainty.

The last class is **imprecise** uncertainty which is introduced due to imprecision. The imprecision can be caused by unclear definition (e.g., ambiguity, vagueness) or the limitation of measurement/assessment techniques. In other words, this uncertainty may expose an object together with the previous uncertainties.

**Example 8.** Analysts should be able to assess the severity of the hand-off failure.

How analysts can come up with the precise value? It is common if the result is just a vague figure. In other situations, the fuzziness is introduced due to the limitation of the techniques.

**Example 9.** To reduce uncertainties of the event in example 6, ones may conduct interviews to some controllers (defined by a statistical sampling technique).

However, the outcome of the interviews still contains the uncertainty (called error) due to the limitation of the techniques. Essentially, the imprecise uncertainty is not the source of risks, but they play a role in defining the extent of risks. In other words, the fuzziness is not a *risk source*, unlike the aleatory and the epistemic ones, but it is a *risk factor*.

In the next section, we propose a framework that incorporates, almost, all uncertainties. We believe incorporating those uncertainties in one framework will result in a better understanding about the complexity of problems, especially in S&D context, besides improving the result of analysis and design.

## IV. Toward a Generic Framework for Secure and Dependable Risks Assessment

A framework is a basic structure that is used to solve or to address complex issues. Ideally, a framework is composed of models, analysis techniques, a process to develop and analyse models, and tools (optional) to assist the framework users. In this paper, we present our revised modelling framework, namely the Goal-Risk framework [27], that has been developed for a general risk analysis. We intend to improve the GR framework to make it more suitable for S&D systems. However, the wisdom of risk management is how to suppress risks that prevents organizations/enterprise in generating values [20]. Our proposal aims also to go in this direction, supporting analysts in managing S&D risks at organization level and considering both socio and technical aspects of a system. By means of the model, analysts and stakeholders are able to collaborate easily in managing risk.

**Example 10.** Controller supervisors suppose to be aware about all possible operational risks concerning system integrity. However, S&D analysts know about the reliability of radar systems, but they are not well informed how the organization (i.e., Air Traffic Service Provider) perceived possible failures, especially in terms of possible cash-flow disturbance or any legal issues that the organization may face.

One may argue that S&D risk is the "usual" risk within an enterprise (e.g., financial risk, legal risk, environmental risk, operational risk, image risk). In our experience, S&D risks are, indeed, alike to these risks. One particularity is the security risk adheres to malicious intent, which rarely is considered in the conventional risk analysis - especially for operational risks. Most of S&D risks are exposed to three types of uncertainty as mentioned before, and they trigger other "usual" risks. Moreover, some S&D risks may not emerge as visible phenomena until an exhaustive audit is conducted (e.g., Enron company scandals) or when they emerge as a result of system functionalities (e.g., John Rusnak scandals [28]). Finally, there is no absolute such secure and dependable system, because it is hardly possible and makes the system so expensive or hard to use. Therefore, the framework should allow analysts to perform a trade-off analysis over several criteria they are interested.

### A. Metamodel for the S&D Risk Model

Surveying different risk management approaches across domain applications (e.g., safety-critical [5], [9], [12], security-critical [29], [30], financial [31], enterprise management [20], [32]) we came up with the metamodel depicted in Fig. 2.

Starting from *risk*, we define it as an uncertain event that produces (negative) impacts to assets [7]. In our framework, we extend the notion of asset beyond resources that are valuable for the organization. In [33], we categorize risks as: risks threaten resources, risks threaten business processes/tasks, and risks obstruct the achievement of strategic objectives/goals. Events are perceived as normal circumstances that occur in a given time and place [34] or failures in a system. Failures might occur due to malicious intents - attacks (e.g., wiretapping) or just accidental actions - accidents (e.g., filling the wrong values). Basically, an attack (or accident) is a threat (or incident) that exploits (or activates) vulnerabilities/faults in the system. If we can fully control (i.e., protect or remove) all the system's vulnerabilities, then we may reach the "absolute" security and dependably [13]. The main differences between a threat and an incident are a threat should have three components: attacker(s)/agent(s), motivation(s), and means(s) while an incident does not necessarily have all those components.

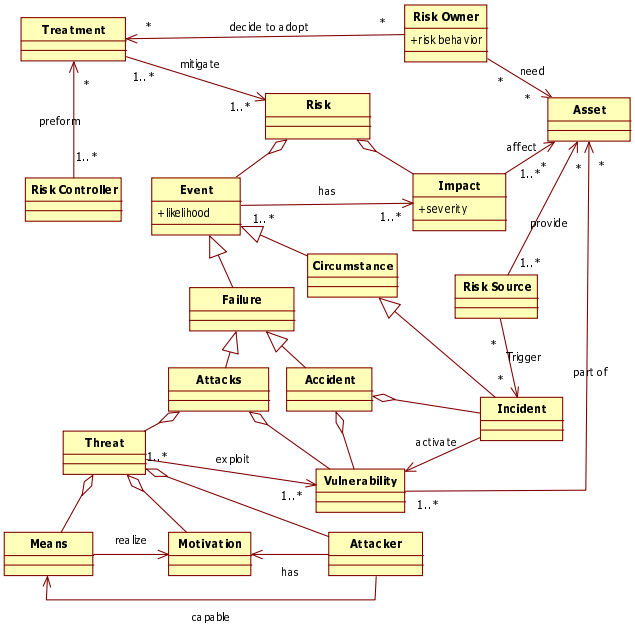


Fig. 2. Metamodel for Secure and Dependable Risk Model

Besides attackers, we consider other three relevant roles: risk owner, risk source, and risk controller [35] and [20]. For a particular risk, these roles are not necessarily played by same actors[3]. Essentially, risk owners are affected actors in case a failure occurs. In other words, the owners may not being the ones that require the assets [37]. Risk sources are actors that provide the assets that contain vulnerabilities (e.g., provider [37]) or actors that triggers vulnerability of the system. Risk controllers are actors that are responsible to mitigate the risk in terms of reducing its likelihood or alleviating its severity.

**Example 11.** Consider the mis-entry risk of an aircraft flight-plan. The *owner* of the risk is air traffic controllers. A flight-data processor (FDP) is the *source* of risk because it is the place where the vulnerability lays. However, Air traffic planners are also the *source* because they are the ones that trigger an incident that result in an accident by activating the vulnerability in the FDP. To reduce the likelihood, supervisors act as the risk *controller* that monitors all new entries flight-plan.

### *B. S&D Risk Modelling Framework*

In this subsection, we refine the metamodel of the Goal-Risk (GR) modelling framework [27], [38] in the light of our findings mentioned in (sub)sections II-B, III, and IV-A. In Fig. 2, we present the mapping for representing the S&D risk assessment concepts using the GR modelling framework in Tab. I.

Unlike other works based on Tropos/*i** [30], [39], [40] that model security as goals, the GR models security and dependability as properties (i.e., reliability, confidentiality, and integrity) of each concept which is needed to be satisfied. In this way, the difference between stakeholders' interests and

[3]Actors [36] includes both human agents and technical systems

| S&D Risk Concepts | GR Constructs |
|---|---|
| Asset | Goal, Task, Resource |
| Vulnerability | Inherent in Goal,Task, Resource |
| Risk | |
| Impact | Impact Relation |
| Event | Event |
| Threat | |
| Attacker | "Bad" Actor |
| Motivation | "Anti" Goal |
| Means | "Malicious" Task |
| Treatment | Task |
| Mitigation | Alleviation or Contribution Relation |
| Risk Owner | "requester" Actor |
| Risk Source | "provider" Actor or "trigger" actor |
| Risk Controller | "controller" Actor |

TABLE I
MAPPING S&D RISK METAMODEL AND GR METAMODEL

S&D properties is more apparent and not because the "label" refers to functionality or S&D.

Essentially, a GR model is composed of three conceptual layers: asset, event, and treatment layer. The asset layer captures "things" that are able to generate values for the actor which are depicted in terms of goals, tasks, and resources. *Goals* (depicted as ovals) represent the objectives that actors intend to achieve. *Tasks* (depicted as hexagons) are course of actions used to achieve goals or treat events. *Resources* (depicted as rectangles) are artefacts that are required to achieve goals or to perform tasks. Moreover, resources can be resulted from the execution of tasks. The event layer depicts uncertain events that can affect the asset layer [4]. The treatment layer represents a set of additional measures, depicted as tasks, to mitigate risks (i.e., their likelihood or impacts).

Initially, we only concentrate on assessing the risk of assets' reliability. It aims at ensuring assets will operate correctly to achieve the top goals of each stakeholders. The scene presented in Example 11 are captured by a GR model as depicted in Fig. 3. In that setting (i.e., the risk of unreliable Flight Plan), Controller is the one that own this risk. Typically, the decision on whether treating risks or not is driven by the owner. In this setting, supervisor acts as risk controller which is responsible to confirm new entries of flight plan by aiming at reducing the likelihood of mis-entry of "null" flight ID. The risk can be triggered by an attack launched by an attacker or simply accidents (e.g., Random Failure FDP Server or Mis-entry of"null" flight ID).

Essentially, both uncertain events activate/exploit the vulnerability at flight plan[5]. Attacks are characterised by attackers, motivation (shutdown Flight Data Processor), and means (Fault Injection). In contrast, accidents happen without any motivations; trigger for normal activities (e.g., mis-entry of "null" flight ID) or random events (e.g., random failures of FDP server, Thunderbolt hit the FDP power grid). Notice there **should not be any "positive contribution"** from the actor's goals/tasks to incidents. Indeed a goal (e.g., working in maximum capacity) can increase the likelihood of an

[4]The GR model allows us to model risks (events with negative impacts) and opportunities (events with positive impacts)

[5]In [41], the authors presented the way to represent explicitly the vulnerability by annotating the resource

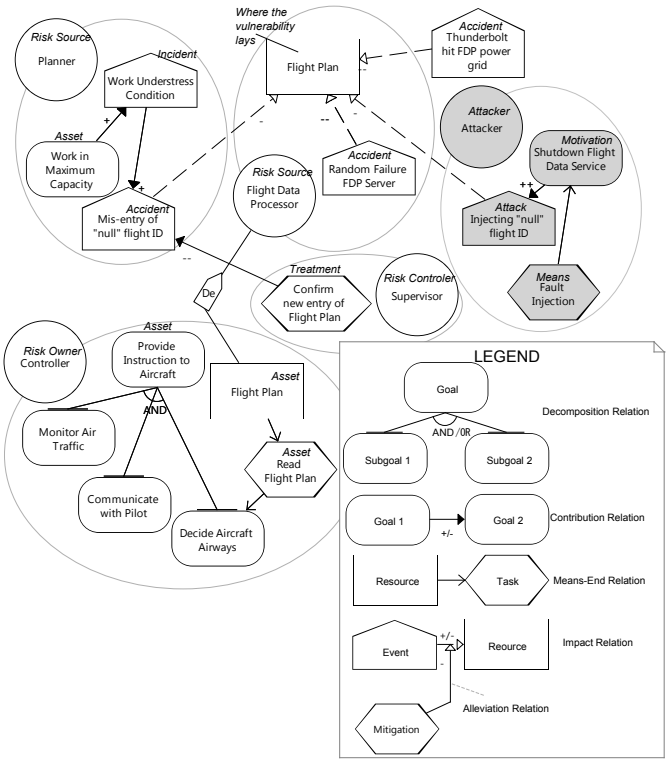


Fig. 3. The GR model in Capturing Reliability Risks

accident, but it is not a direct contribution. There must be an intermediate event (in ATM domain is called by incident) in between. As far as there is a mechanism that suppresses the occurrence of the incident, then the goal achievement will not increase the accident likelihood.

In the context of assessing confidentiality and integrity, we use a similar approach taken for the reliability. Those attributes are represented as a tag (e.g., "I" or "C") for the asset layer as depicted in Fig. 3. Unlike reliability, not all assets are confidential/integrity-sensitive. This tagging process can be done by stakeholders and later refined by analysts. Likewise Impact relation, an integrity impact is depicted using "dash-line ending by box", and a confidential impact is depicted by "dash-line ending by diamond". Both relations also indicate their severity with a label $(-,--)$.

Confidentiality risks only threaten resources (not goals and tasks) while integrity risks can expose all types of assets. Moreover, confidentiality are prone to risks resulted from attacks, while an integrity breach can be resulted from the normal operation.

**Example 12.** The event mistype flight ID may result in the breach of the flight data integrity. However, the event might be caused by a normal operation of a planner who is careless.

By means of this model, ones can model S&D risks particularly reliable, confidential, and integrity. Each quality is characterised by the likelihood and total expected loss[6]. However, those three qualities are exposed to the three types of

[6]Notice that each assets has a value. Loss is the sum-up of the loss of value.

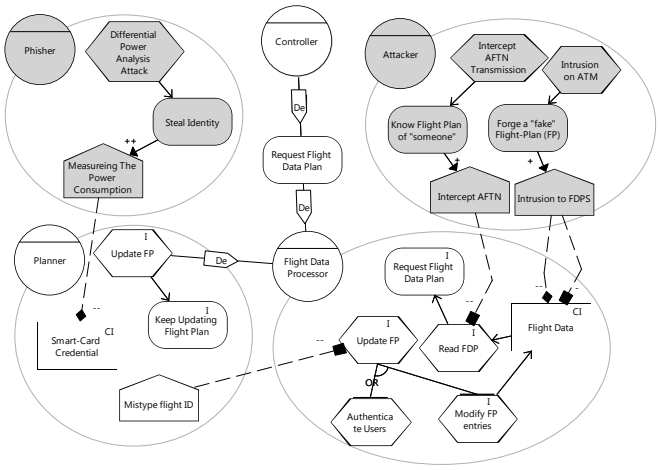


Fig. 4. The GR model in Capturing Confidential and Integrity Risk

uncertainty explained in Section III. To represent such uncertainties, each constructs are quantified in terms of its evidence (supporting, opposing) to be correctly operated adapting from the *Dempster-Shafer Theory of Evidence*.

$$Sat_i + Den_i + X_i + Y_i = 1; i \in Graph\_nodes^{7}$$

For the complete calculus of this mathematical model, readers may read [42]. In contrast with *Probability Theory*, here the opposing evidence (denial) cannot be calculated from the supporting evidence (satisfaction) (i.e., $den_i \neq 1 - sat_i$). Moreover, we model the *lack of knowledge-$X_i$* and *conflicting of knowledge-$Y_i$* that are required for epistemic uncertainty.

If an event is only exposed to the aleatory nature, we can assume that there is no lack and conflicting knowledge (i.e., $den_i = 1 - sat_i$). For epistemic uncertainty, analysts do not have complete knowledge about the event therefore $sat_i + den_i \leq 1$ and the reminder $(1 - sat_i - den_i = X_i)$ can be perceived as the lack of knowledge. By means of the automated reasoner (adapted from [19], [38]), the model obtains the final evidence values. Principally, each relation in the GR model propagates the evidence from source nodes to final nodes following defined semantics [38], [42]. At final, each node has the final evidence for each attributes.

**Example 13.** Based on Fig. 3, initially the attack injecting "null" flight ID has 50% sat evidence that it will be launched. At the end, the attack injecting "null" flight ID has sat $= 0.4$, den $= 0.5$, and $Y = 0.1$.

Notice at initial analysts assesses the sat only and not talking about denial of the attack. Therefore, ones may assume that the analysts do not have any information for the reminder $(X = 0.5)$. The example 13 indicates that at the end the sat is lower than the initial one. It might be due to the fact that the attacker is not "really" motivated to shutdown flight data service (den=0.6). Since it is impossible to have the sum of evidence bigger than 1, one may assume there is conflicting evidence $(Y = 0.1)$.

[7]$Sat_i \geq 0; Den_i \geq 0; X_i \geq 0; Y_i \geq 0$

Based on those final values, ones may compute how much the likelihood ($\lambda$) of "things" occurs (i.e., goals to be achieved, tasks to be executed, resources to be provided, and events occur). In this framework, if an event is exposed only by aleatory then $\lambda = \mathsf{sat}$. If events (and also goals, tasks, resources) are exposed in epistemic one (i.e., $X \neq 0$ or $Y \neq 0$) then the likelihood is a imprecise value. In this framework, we define it as a range:

$$bel_i \leq \lambda_i \leq pla_i{}^{8}; i \in Graph\_nodes$$

$$bel_i = \begin{cases} Sat_i, & \text{if } i \text{ is a risk/bad thing;} \\ Sat_i - [\![X_i - Y_i]\!], & \text{otherwise}^{9}. \end{cases}$$

$$pla_i = \begin{cases} Sat_i + X_i + Y_i, & \text{if } i \text{ is a risk/bad thing;} \\ Sat_i, & \text{otherwise.} \end{cases}$$

Essentially, the belief $bel_i$ assumes at the "at least" value for evidence while the plausibility $pla_i$ assumes at the "most" value one. By means this representation, ones can see the (un)fuzziness of the evidence of a goal and decide how much mitigation are required accordingly.

By means of the model and its formalization, analysts can understand what are risks associated to the system and assess them. Moreover, they can also understand how to reduce the uncertainty, according to its nature, of risk and how to mitigate the risk [27]. For instance, if a risk (or an opportunity) has a wide-range (imprecise) of likelihood (e.g., 20%-70%) then it is better if the treatment is elicited based on the biggest (smallest) value. In the case of incomplete knowledge of an event, analysts can perform some empirical efforts (e.g., interviews, expert judgements) to reduce the lack of knowledge or to minimise the conflict of knowledge about the event. Consequently, these efforts may reduce the uncertainty about the event. Imprecise uncertainty is the only type that may not result in risk. It depends on the sensitivity of the system towards the imprecision aspect.

## V. Concluding Remarks

In this paper, we have discussed about basic concepts of security and dependability proposed in literature and how they are related to uncertainty. We have explained how S&D quality attributes can be reduced to reliability, confidentiality, integrity, and recoverability qualities. We have proposed a modelling framework that captures critical concepts of having a S&D system (e.g., assets, attack, accident, failures, treatment) and reason about related risk from three different dimensions (namely, reliability, confidentiality, and integrity). The nature of uncertainty is used along the assessment process and is used to find the most appropriate treatment for risk mitigation. Moreover, one may perform a cost-benefit analysis [44] over a GR model to maximise the effectiveness of cost for countermeasures in achieving those three attributes using the forward reasoning [38]. For enrichment, trade-off analysis [41] can be conducted using a GR model to see the trade-off among attributes for a given set of treatments.

Currently, our approach is able to capture and analyse some parts of availability property (i.e., reliability, recoverability) [33]. However, it cannot deal with the interruption (e.g., Denial-of-Service) because in this case the system runs correctly (reliable) but it is not accessible by end-users. Our model elicits "necessary" evidence for the availability of a system, but further analyses (i.e., responsiveness) are required to ensure the availability. In this paper, we model and analyse necessary conditions to be secure and dependable, but unfortunately it is not, surely, sufficient (i.e., if there both conditions are different). In fact, it is hardly possible to have sufficient evidence/condition to judge whether the system is secure or *not*. Security engineering, often, conducts with lack of knowledge. For instance, our firewall can filter all malicious traffic now, but not necessarily the case for next week. It is due to the nature of the quality attributes which their breaches are not necessarily appeared by users and always increase over the time.

The value of $X$ or $Y$ refers to the incomplete knowledge of the analysts about the subject. Ideally, they should try to reduce those values using some techniques as indicated in the Section III. However, those techniques cannot be counted as mitigation. It is aiming at increasing the preciseness of the assessment, and at the end it may reduce the possibility of "over-shoot" in adopting countermeasures (i.e., reduce the cost of countermeasures).

Analysts should try to remove the vulnerability that lay in the system. In [3], [37], [45], the authors proposed structure ways to identify vulnerabilities and possible removals. However, we realise it is impossible to remove all vulnerabilities. Therefore, a set of means is necessary to mitigate risks: prevention, detection-recovery, alleviation, and restoration. Prevention means aims at preventing the exploitation/activation of those vulnerabilities. Detection-recovery means aim at detecting the present of intrusions or errors and try to recover the system from them. Alleviation intends to reduce the severity of failures resulted from attack or accident. Finally, analysts should provide means to perform restoration after failures. Here, we distinguish between restoration and recovery. Restoration recovers the system from failures while recovery is aiming at recovering from errors. However, in many cases they are alike. In some setting, we may transfer the risks into the third party (e.g., insurance companies) which can be seen as part of the mitigation strategies.

As mention in the previous section, risk controller and risk owner can be different actors. It might be the case that the controller does not perceive a risk as greater as the owner. This situation may result in agreed countermeasure are not executed because the controller perceives less risky than the one that owner does. Therefore, further analyses are required to ensure that the perceived risks do not significantly deviate from the actual ones.

[8] It is specified following the similar notion on **belief** and **plausibility** in the Dempster-Shafer Theory [43]

[9] $[\![x]\!] = X$ if $x \geq 0$, otherwise $[\![x]\!] = 0$

## ACKNOWLEDGEMENTS

This work was supported by funds from the European Commission through EU-SERENITY Project.

## REFERENCES

[1] A. Avizienis, J.-C. Laprie, B. Randell, and C. E. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *TDSC*, vol. 1, no. 1, pp. 11–33, 2004. [Online]. Available: http://csdl.computer.org/comp/trans/tq/2004/01/q0011abs.htm

[2] E. Jonsson, "Towards an Integrated Conceptual Model of Security and Dependability," in *Proc. of ARES'06*. Los Alamitos, CA, USA: IEEE CS Press, 2006, pp. 646–653.

[3] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A Taxonomy of Computer Program Security Flaws," *ACM Comp. Surveys*, vol. 26, no. 3, pp. 211–254, 1994.

[4] M. J. Carr, S. L. Konda, I. Monarch, F. C. Ulrich, and C. F. Walker, "Taxonomy-Based Risk Identification," SEI-CMU, Tech. Rep. CMU/SEI-93-TR-6, June 1993.

[5] M. Stamatelatos, W. Vesely, J. Dugan, J. Fragola, J. Minarick, and J. Railsback, *Fault Tree Handbook with Aerospace Applications*. NASA, 2002.

[6] I. Sommerville, *Software Engineering*, 7th ed. Addison Wesley, May 2004.

[7] ISO/IEC, "Risk Management-Vocabulary-Guidelines for Use in Standards," ISO/IEC Guide 73, ISO/IEC, 2002.

[8] B. Schneier, "Attack Trees: Modeling Security Threats," *Dr. Dobbïs Journal*, vol. 12, no. 24, pp. 21–29, 1999.

[9] T. Bedford and R. Cooke, *Probabilistic Risk Analysis: Foundations and Methods*. Cambridge University Press, 2001.

[10] R. Yager, "On the Dempster-Shafer Framework and New Combination Rules," *Information Sciences*, vol. 41, no. 2, pp. 93–137, 1987.

[11] US DoD, "Trusted Computer System Evaluation Criteria," DoD 5200.28-STD, December 1985.

[12] US-DoD, "Military Standard, Procedures for Performing a Failure Mode, Effects, and Critical Analysis," MIL-STD-1629A, 1980.

[13] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 4th ed. Prentice-Hall, 2006.

[14] D. Bodeaum, "A Conceptual Model for Computer Security Risk Analysis," in *Proc. of 8th ACSAC*, San Antonio, TX, USA, 1992, pp. 56–63.

[15] ISO, "Information technology - security techniques - code of practice for information security management," ISO 17799, ISO, 2005.

[16] ISO/IEC, "Management of Information and Communication Technology Security - Part 1: Concepts and Models for Information and Communication Technology Security Management," ISO/IEC 13335, ISO/IEC, 2004.

[17] J. Walkerdine, L. Melville, and I. Sommerville, "Dependability Properties of P2P Architectures," in *Proceedings. Second International Conference on Peer-to-Peer Computing (P2P 2002)*. IEEE CS Press, 2002, pp. 173–174.

[18] J. Pearl, "Bayesianism and Causality, or, Why I am Only a Half-Bayesian," in *Foundations of Bayesianism*, ser. Applied Logic. the Netherlands: Kluwer Academic Publishers, 2001, vol. 24, pp. 19–36.

[19] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Formal Reasoning Techniques for Goal Models," *Journal of Data Semantics*, vol. 1, no. 1, pp. 1–20, October 2003. [Online]. Available: http://dit.unitn.it/~pgiorgio/papers/jdatasemantics-2004.pdf

[20] COSO, *Enterprise Risk Management - Integrated Framework*, http://www.coso.org/Publications/ERM/COSO_ERM_ExecutiveSummary.pdf, Committee of Sponsoring Organizations of the Treadway Commission, September 2004. [Online]. Available: http://www.coso.org/Publications/ERM/COSO_ERM_ExecutiveSummary.pdf

[21] D. V. Lindley, *Understanding Uncertainty*. John Wiley & Sons, 1923.

[22] M. Smithson, *Ignorance and Uncertainty: Emerging Paradigms*. Springer, 1989.

[23] A. O'Hagan, C. E. Buck, A. Daneshkhah, J. R. Eiser, P. H. Garthwaite, D. J. Jenkinson, J. E. Oakley, and T. Rakow, *Uncertain Judgements: Eliciting Experts' Probabilities*. Wiley, 2006.

[24] M. van Staveren, *Uncertainty and Ground Conditions: A Risk Management Approach*. Elsevier, 2006.

[25] R. Hanmer, *Patterns for Fault Tolerant Software*. Wiley, Dec. 2007.

[26] P. Kocher, J. Jaffe, and B. Jun, "Introduction to Differential Power Analysis and Related Attacks," http://www.cryptography.com/technology/dpa/DPATechnicalInfo.PDF, 1998.

[27] Y. Asnar and P. Giorgini, "Modelling Risk and Identifying Countermeasures in Organizations," in *Proc. of CRITIS'06*, ser. LNCS, vol. 4347. Springer, 2006, pp. 55–66. [Online]. Available: http://yudis.asnar.googlepages.com/asnar-girogini-critis2006.pdf

[28] F. Massacci and N. Zannone, "Detecting Conflicts between Functional and Security Requirements with Secure Tropos: John Rusnak and the Allied Irish Bank," in *Social Modeling for Requirements Engineering*. MIT Press, 2007, to appear.

[29] CORAS, "CORAS: A Platform for Risk Analysis of Security Critical System," http://www.nr.no/coras/, 2005, accessed at September 2005.

[30] N. Mayer, E. Dobuis, and A. Rifaut, "Requirements Engineering for Improving Business/IT Alignment in Security Risk Management Methods," in *Proc. of I-ESA'07*, 2007.

[31] D. Vose, *Risk Analysis: A Quantitative Guide*. Wiley, 2000.

[32] J. King, *Operational Risk: Measurement and Modelling*. Wiley, 2001.

[33] Y. Asnar and P. Giorgini, "Analyzing Business Continuity through a Multi-Layers Modell," in *Proc. of 6th Int. Conf. on Business Process Management*, 2008.

[34] WordNet, "WordNet - a lexical database for the English language," http://wordnet.princeton.edu/, 2005, (access on November 2005). [Online]. Available: http://wordnet.princeton.edu/

[35] The IT Governance Institute, "Framework control objectives management guidelines maturity model," 2007.

[36] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An Agent-Oriented Software Development Methodology," *JAAMAS*, vol. 8, no. 3, pp. 203–236, 2004.

[37] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Requirements Engineering for Trust Management: Model, Methodology, and Reasoning," *Int. J. of Inform. Sec.*, vol. 5, no. 4, pp. 257–274, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10207-006-0005-7

[38] Y. Asnar and P. Giorgini, "Risk Analysis as part of the Requirements Engineering Process," DIT - University of Trento, Tech. Rep. DIT-07-014, March 2007. [Online]. Available: http://eprints.biblio.unitn.it/archive/00001180/01/QualitativeGR-tech-rep.pdf

[39] L. Liu, E. S. K. Yu, and J. Mylopoulos, "Security and Privacy Requirements Analysis within a Social Setting," in *Proc. of RE'03*, 2003, pp. 151–161. [Online]. Available: http://csdl.computer.org/comp/proceedings/re/2003/1980/00/19800151abs.htm

[40] H. Mouratidis and P. Giorgini, "Secure Tropos: Dealing effectively with Security Requirements in the development of Multiagent Systems," in *Safety and Security in Multiagent Systems*, ser. LNCS. Springer, 2006.

[41] G. Elahi and E. Yu, "A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs," in *Proc. of ER'07*, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75563-0_26

[42] Y. Asnar and P. Giorgini, "Analysing Risk-Countermeasure in Organizations: a Quantitative Approach," DIT - University of Trento, Tech. Rep. DIT-07-047, July 2007.

[43] G. Shafer, *The Dempster-Shafer Theory*. Wiley, 1992, pp. 330–331.

[44] J. Willemson, "On the gordon & loeb model for information security investment," in *Proceedings of The Fifth Workshop on the Economics of Information Security (WEIS '06)*, 2006.

[45] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Detecting conflicts of interest," in *Proc. of RE'06*. Los Alamitos, CA, USA: IEEE CS Press, 2006, pp. 308–311.