UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

Analyzing Risk-Countermeasure in Organizations: a Quantitative Approach

Yudistira Asnar (yudis.asnar@dit.unitn.it)
Paolo Giorgini (paolo.giorgini@dit.unitn.it)

July 2007

Technical Report # DIT-07-047

Analyzing Risk-Countermeasure in Organizations: a Quantitative Approach

Abstract

Risk is one of inherent problems in all software systems. It becomes more significant if the software system is operated in a critical system (e.g., air traffic control, nuclear plant). It is because in this domain the software system is expected to be always dependable all the time of its operation. The system is dependable when all its risks are suppressed until acceptable level. Therefore, in such setting analysts must carefully analyze the socio-technical system (i.e., organizational-setting and software systems) and understand how uncertain events may affect the systems. By means of the Tropos Goal-Risk, we model the socio-technical system including its risks. Essentially, the framework consists of goal, event, and treatment modeling. The goal layer represents what the stakeholders' interests are and how to achieve them. The event layer depicts how uncertain events occur and impact the goals of stakeholders. The treatment layer represents what the possible measures that are available to treat the events. By quantifying the evidence value of the model, analysts can reason about the level of risk and choose the most appropriate alternative to achieve the stakeholders' interests and the necessary treatment that should be employed to mitigate the risks. We use a case study on Air Traffic Management to illustrate the proposal.

1 Introduction

Software systems are more and more part of our daily life, and very often they have a strong influence in our daily life decisions. In this setting, a software system plays a critical roles. In literature [36], one distinguishes a critical system into: safety-critical system (its failure may result in the loss of life or the environment damage direct or indirectly), mission-critical system (its failure may cause to the failure of any activities that are means to achieve the organization goals), and business-critical system (its failure may result to the high economic losses).

In such scenario, a model plays a main role to communicate the stakeholders' mind and modelers/analysts about the stakeholders' interests. It is important because stakeholders are the ones that really understand how the organization operates and acquire full knowledge about the domain application of the software system. However, stakeholders, typically, lack of technical expertises to realize the software system. Thus, they need developers (i.e., analysts, designers, programmers) to realize their system. In fact, many software systems fail to operate as their intended purposes because of

mis-understanding between stakeholders and developers (especially analysts) [43]. It happens, mainly, during a requirement modeling phase when the strategic-interests of stakeholders are identified. Ideally, a model should have a precise semantic definition for each construct. So that, each actor, which is involved in the modeling process, has the same interpretation for the same model of the system. By means of a conceptual model, analysts may verify their understanding about the existing system or the stakeholders' intentions. All of these will reduce the likelihood of having false or unnecessary requirements.

In requirement engineering community, Goal-oriented RE methodologies and frameworks (e.g., KAOS [10], *i** [44], GBRAM [1], and Tropos [6]) emerge as a research area where the concept of goal is used to facilitate analysts understanding the strategic-interests of stakeholders and then motivates the system requirements within the organizational setting. Particularly, Tropos, which uses the *i** as modeling framework, proposes an early requirements analysis phase. During this phase, analysts learn about the problems by studying the organizational-setting where the system will operate. This phase results an organizational model which consists of relevant stakeholders or actors, their goals, and their interdependencies (i.e., who depend on one another for goals to be fulfilled, tasks to be performed, and resources to be furnished). Through these dependencies, one can answer *why* questions, besides *what* and *how*, regarding system functionalities or requirements. Answers to *why* questions, ultimately, link system functionalities to stakeholders' interests, preferences, and intentions.

Though there are several attempts [25, 17] to enrich the Tropos/*i** modeling framework to model a critical software systems. However, we still found them inadequate in depicting how a failure is developed or an attack occurs. This understanding about failures or attacks is necessary to analyze alternative requirements and, if it is necessary, to introduce additional mechanisms to mitigate the risks. In this work, we uses Tropos Goal-Risk (GR) framework [3] to fill this gap by modeling them (i.e., attacks, failures) as the event layer of the Tropos GR framework. We will explain, in detail, the meta-model of the event layer, such that it is able to model failures and attacks naturally. Based on this model, ones may elicit necessary countermeasures to mitigate the risks which are introduced by the event layer.

The rest of the paper is organized as follows: We briefly explain about Air Traffic Management (§2) which is used to explain how to model the current situation using a Risk-Countermeasure framework, namely Tropos Goal-Risk framework (§3) and later analyzed it. Essentially, Tropos GR framework consists in three layers: goal, event, and treatment layers. The goal layer represents what the stakeholders' interests are and how to achieve them. The event layer depicts how uncertain events occur and impact the goals of stakeholders. The treatment layer represents what possible measures are available to treat the events. Our contribution, mainly, is in proposing a quantitative model of the Tropos GR framework which based on the Dempster-Shafer theory of evidence [34]. Later, we present the related work for our proposal (§4). Finally, we conclude the paper with a final discussion and future works (§5).

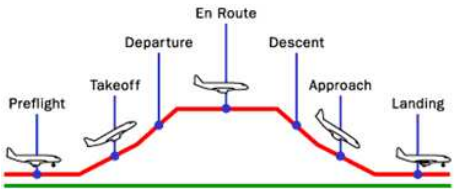


Figure 1: Profile of a Commercial Flight

2 Scenario

Essentially, Air Traffic Management (ATM) [22] is categorized as a critical socio-technical systems because its operations highly involves human interaction, and its failures may result human-life losses. Thereby, it is required to be dependable during all its operation. Based on [15], ATM is an aggregation of services that consists of Airspace Organization and Management (AO&M), Air Traffic Flow and Capacity Management (ATCFM), Air Traffic Control (ATC), Airport operations, Aircraft operations, and Information management. In this section, we briefly explain ATC system which is mainly responsible to maintain a safe, orderly, and efficient flow of traffic. This service is provided by ground-based Air Traffic Controllers (ATCOs) to maintain safe operation of aircraft and any related third parties (e.g., ground-crews, airports). Such as, ATCOs should control aircraft to maintain horizontal and vertical separation among aircraft in their airspace. Moreover, they must ensure traffic flows orderly and provide flight context information to pilots (e.g., routes, weather conditions, etc.).

These services are organized by an authorized body depending on the phase of a flight (e.g., pre-flight, take-off, departure, en-route, descent, approach, landing). During pre-flight and take-off phases, ATC services are provided by ATCOs operating at the departure airport. In case for approach and landing phases, an aircraft is controlled by ATCO's which operates in the tower on arrival airport. Climb, en-route, and approach phases are managed by ATCOs operating in the various Air-traffic Control Centers (ACCs) where the aircraft passes through. All operations of ATCOs are defined in very elaborate procedures driven by imperative safety requirements and regulated by air navigation authorities (e.g., EUROCONTROL, FAA, ICAO).

Several accidents, such as: aircraft collision, controlled flight into terrain, and wake turbulence, are still becoming main concerns of many air navigation authorities. Based on [14], aircraft collision can be distinguished into three classes: mid-air collision (while an aircraft is airborne), runway collision (while an aircraft in runway), and taxiway collision (while an aircraft in taxiway). Controlled flight into terrain (CFIT) is an accident in which aircraft, under the control of the crew, is flown into terrain (or water, obstacle) with no prior awareness on the part of the crew of the impending accident. Wake turbulence is an accident in which an aircraft gets severe effect of the rotating air masses generated behind the wing tips of a large jet aircraft. In this paper, we present the use of a conceptual model, Tropos Goal-Risk in particular, to model how such accidents are occurred and impact the business goal of ATC services using the data [14, 37] from EUROCONTROL.

3 Tropos Goal-Risk Framework

Tropos is a software development methodology that adopts the concept of agent and its related mentalistic notions (e.g., goals, tasks, and resources) along all the phases of development process [6]. The methodology spans from early requirements analysis up to implementation. It uses goal models to represents agent (or more general actor) mental states [19]. The key role of early requirements analysis is to model the stakeholders' strategic-interests and the system-to-be, together with the organizational-setting where the system will operate.

This modeling framework seems to be inadequate to model such a critical system where the failure of a system causes severely consequences to the stakeholders (even to the environment). Therefore, Goal-Risk (GR) framework [3] is introduced extending the Tropos goal model [18, 33] by introducing constructs and relations specific for analyzing risk. So it can model any circumstances that cause failures and deals with them. By means of this framework, an analyst may analyze the system model, in terms of its risks and possibly introduces any countermeasures to mitigate unacceptable risks.

In this paper, we propose a quantitative model for Tropos GR model. Thus, it facilitate analysts to model and reason about risk of the stakeholders' interests using quantitative data. This quantitative model is developed based on the Dempster-Shafer theory of evidence [34]. This theory allows us to operate using subjective data, besides only the objective one. Moreover, the Tropos GR model promotes a better understanding about:

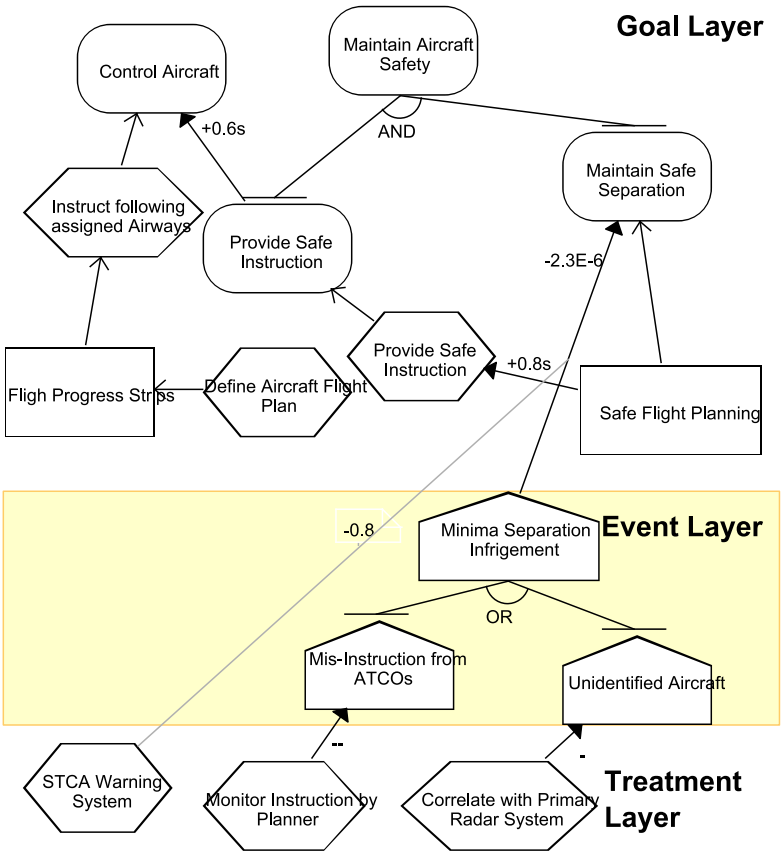
- what the interests (i.e., desires, intentions) of the stakeholders are;
- what the risks of the interests are, and how they are developed/occurred;

By this understanding, analysts may assess the risk of the stakeholders' interests (and consequently the system-to-be) and reacts accordingly (e.g., introducing any countermeasures, eliminating such requirements). From now on, the term actor and stakeholder are used interchangeable throughout this paper.

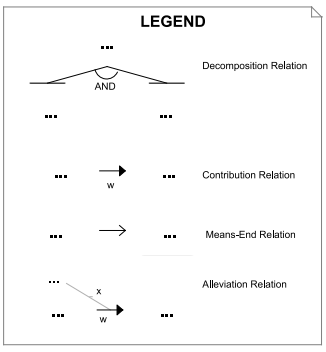
Essentially, a Goal-Risk (GR) model is represented as a graph $\langle \mathcal{N}, \mathcal{R}, \mathcal{I} \rangle$, where $\mathcal{N}$ are nodes and $\mathcal{R}$ are relations (see Fig. 2). $\mathcal{N}$ is comprised of *goals*, *tasks*, *resources*, and *events*. These nodes are interrelated among them with any relations in $\mathcal{R}$ which are detailed in the modeling subsections. Goal (depicted as oval) is a strategic-interest that an actor intends to achieve. Task (depicted as hexagon) are a sequence of actions used to achieve a goal or to treat an event. Resource (depicted as rectangle) represents a physical or an information entity which can be means to fulfill a goal or needed to execute a task. Event (depicted as pentagon) is uncertain circumstance which is, typically, out of the control of actor that can have an impact (positively or negatively) on the fulfillment of a goal.

Generally, those nodes have two attributes SAT and DEN. These attribute represent the evidence value of a node to be satisfied (i.e., goal to be fulfilled, task to be executed, resource to be furnished, and event to be occurred) and denied respectively. In the probability theory, if $Prob(A) = 0.1$ than we can infer that the probability of $\neg A$ is 0.9 (i.e., $P(\neg A) = 1 - P(A)$). Conversely, following the idea of Dempster-Shafer theory [34] the evidence of the goal[1] being denied (DEN) can not be inferred from the

[1] it is also applied for task, resource, and event



(a)



(b)

Figure 2: Simple Model of Goal-Risk Framework

satisfaction evidence (SAT) and vice versa. Those attributes must fulfill the following

rules:

$$0 \leq Sat(N_i), Den(N_i) \leq 1 \quad (1)$$

$$Sat(N_i) + Den(N_i) + X(N_i) = 1 \quad (2)$$

(1) states that SAT and DEN are positive values which are laid between $[0 \ldots 1]$, and (2) rules that the total of evidence value of N_i must be equal 1. $X(N_i)$ represents the lack of evidence value for deciding the satisfaction or denial of N_i (i.e., ignorance [41]). SAT, somehow, can be seen as the belief-degree [26] of the satisfaction of a node, and conversely for DEN. Later, we will present how to obtain a probability value (i.e., subjective probability) from evidence values.

As mentioned before, the GR model is composed into three layers conceptual analysis: goal layer, event layer, and treatment layer. **Goal layer** analyzes stakeholders' goals in a organization-setting. It also depicts how the stakeholders achieve their goals by means of tasks and resources. For the sake simplicity (i.e., avoiding dependency relationships), the GR model, presented in this paper, assumes all strategic-interests belong to a single actor (called ATC provider). **Event layer** captures all significant uncertain events that may effect the goal layer (mainly the one with negative effect). Risk is defined as the combination of the probability an event and its consequences [23], while in [9] risk is an uncertain events with negative impact which can prevent value creation or erode existing value. Based on the latter definition, the event layer may also capture an event with positive impact (called an opportunity) or, even, the one with both polarities of impact. This feature is useful when the stakeholders start to decide how far they need to suppress the risk with considering the possibility of losing some opportunities. **Treatment layer** represents a set of possible measures that can be introduced to treat the risk of the system. A treatment, in principle, is a special tasks which is operated by reducing the likelihood or reducing the severity of an event. The details of each layer are explained in the following subsections using the ATM scenario.

By means of a GR model, analysts can assess the risk of the system with following the analysis process defined in [3]. The process requires the evidence values of terminal nodes[2] in $\langle \mathcal{N}, \mathcal{R}, \mathcal{I} \rangle$ and, later, the process propagates those values calculating the evidence values of the stakeholders' goals either to be satisfied or to be denied. Analysts may specify certain criteria (e.g., maximum risk level, maximum total costs) to evaluate alternative solutions. In this setting, the process will enumerate all possible alternative solutions, which satisfy given criteria, to achieve stakeholders' goals. Moreover, the event layer depicts how risks are developed (i.e., from terminal-events until top-events) and, finally, impact stakeholders' goals. In [2], we have defined guidelines to define countermeasures which are categorized into 5 types (e.g., avoidance, prevention, detection, alleviation, and retention) based on the structure of the event layer.

3.1 Goal Modeling

GR modeling starts from modeling the goal layer. It models strategic-interests of the stakeholders and how they are fulfilled. Initially, 1) analysts collect the information to identify the stakeholders' strategic-interests. In the ATM scenario (in Fig. 3), an

[2]Vertex that has the evidence values initially. Typically, it does not have any incoming edges

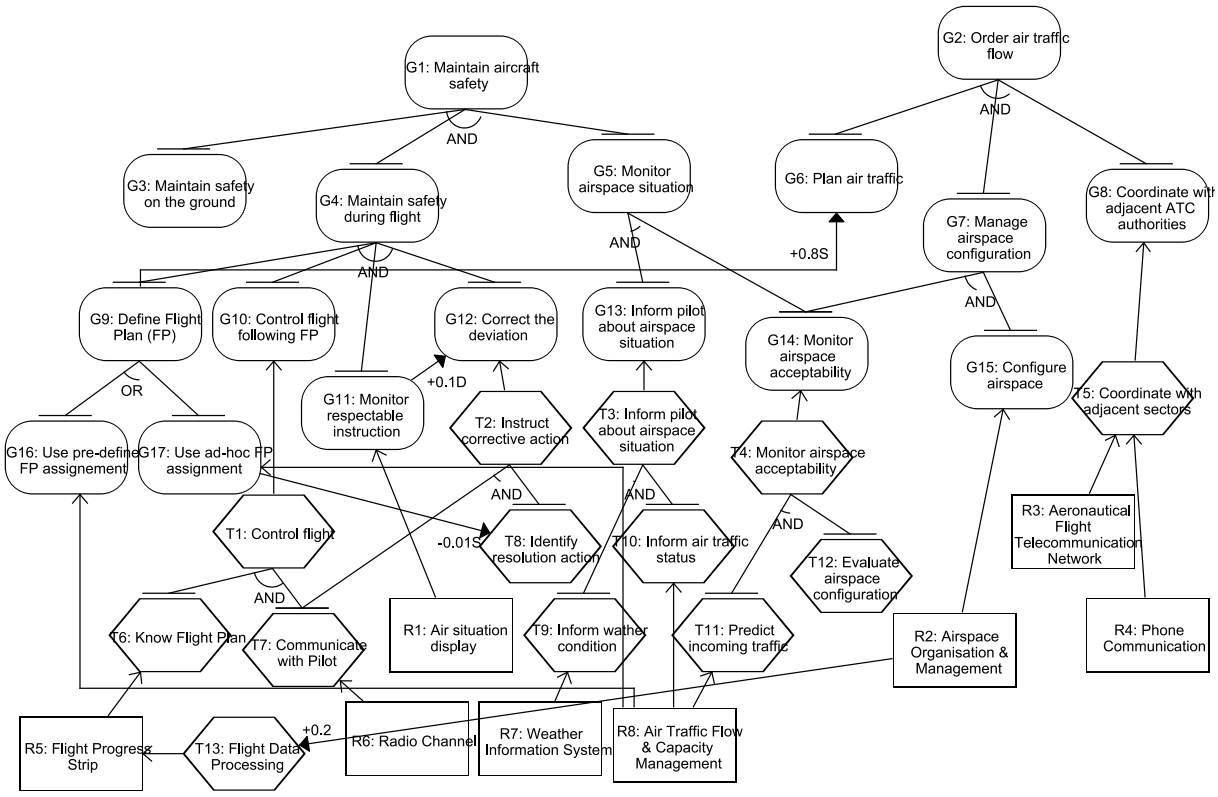


Figure 3: Goal Layer of ATM Scenario

ATC provider primarily aims to maintain aircraft safety *(G_1)* and order air traffic flow *(G_2)* efficiently (called top-level goals). 2) Top-level goals must be decomposed using *AND/OR-decomposition* until all leaf-goals are tangible (i.e., there is an actor that can fulfilled the goals). For the sake of simplicity, we assume that the ATC provider can fulfill its goals without any dependency to other actors. 3) Performs means-end analysis to identify tasks or resources that are means to achieve leaf-goals, and 4) relates among goals, tasks, or resource, if their satisfaction/denial affect the others using contribution relations.

As depicted in Fig. 3, Goal G_1 is refined (*AND-decomposition*) into several subgoals: maintain safety on the ground *(G_3)*, maintain safety during flight *(G_4)*, and monitor airspace situation *(G_5)*. These subgoals must be fulfilled in order to achieve the up-level goal (e.g., G_1). Therefore, in terms of the evidence value of up-level goal are calculated using (3)-(5) which are the adaptation from Yager's ruls [42]. Suppose that G_i and G_j are *AND-subgoals* of G_{up} . Because the up-level goal is fulfilled when both subgoals are fulfilled[3], (3) specifies that the SAT of up-level goal is calculated on the basis of the production of all the SAT of its subgoals. Conversely, the

[3]The framework has not consider the order of the fulfillment of all subgoals yet

DEN of up-level goal is defined in (4) because the up-level goal is failed when there is, at least, one subgoal is failed . The up-level goal is undecidable if there is, at least, an undecidable subgoal and no subgoal is failed, therefore the $X(G_{up})$ is defined in (5);

$$
\begin{aligned}
Sat(G_{up}) &= Sat(G_i) \times Sat(G_j) && (3)\\
Den(G_{up}) &= Den(G_i) \times Sat(G_j) + Den(G_i) \times Den(G_j) + \\
&\quad Den(G_i) \times X(G_j) + \\
&\quad Sat(G_i) \times Den(G_j) + X(G_i) \times Den(G_i) && (4)\\
X(G_{up}) &= Sat(G_i) \times X(G_j) + X(G_i) \times Sat(G_j) + \\
&\quad X(G_i) \times X(G_j) && (5)
\end{aligned}
$$

Moreover, a goal may have several alternatives of its fulfillment (i.e., *OR-decomposition*). For instance, the goal of define Flight Plan (FP) *(G_9)* can be done either by use predefined FP assignment *(G_{16})* where FPs is assigned for particular routes statically, or use ad-hoc FP assignment *(G_{17})* where each flight requests its FP before departure. The fulfillment of either G_{16} or G_{17} can be counted as the fulfillment G_9 . Adopting from the disjunctive consensus rule proposed by Dubois and Prade in [13], we define how to calculate the evidence values of the up-level goal from its subgoals as (6)-(8).

$$
\begin{aligned}
Sat(G_{up}) &= Sat(G_i) \times Sat(G_j) + Sat(G_i) \times Den(G_j) + \\
&\quad Sat(G_i) \times X(G_j) + \\
&\quad Den(G_i) \times Sat(G_j) + X(G_i) \times Sat(G_i) && (6)\\
Den(G_{up}) &= Den(G_i) \times Den(G_j) && (7)\\
X(G_{up}) &= Den(G_i) \times X(G_j) + X(G_i) \times Den(G_j) + \\
&\quad X(G_i) \times X(G_j) && (8)
\end{aligned}
$$

Actually, (6)-(8) mirror the ones from *AND-decomposition*. It is because the SAT evidence is the summation of any situations where, at least, a subgoal is satisfied, and the DEN evidence is calculated when all subgoals are denied. Moreover, both set-of-formalizations (i.e., (3)-(5) and (6)-(8)) satisfy the basic rules (1)(2). An analyst must ensure the evidence from all subgoals are disjoint, otherwise the same evidence are calculate twice.

Tasks and *resources* are introduced as means to achieve leaf-goals in the model. This analysis is depicted using a means-end relation. In Fig. 3, the goal coordinate with adjacent ATC authorities *(G_8)* is fulfilled by executing the task coordinate with adjacent sectors *(T_5)*, and the provision of air space display *(R_1)* allows the ATC authority to achieve the goal monitor respectable instructions *(G_{11})*. A means-end relations may also use to represent which *resources* are needed for *task* executions, like Aeronautical Fixed Telecommunication Network (AFTN) *(R_3)* or phone communication *(R_4)* can be used as a mean to coordinate with adjacent sectors *(T_5)*. In another context, a means-end relation can model the provision of a *resource* by a *task*. For instance, flight data processing *(T_{13})* provides flight progress strips *(R_5)* that,

furtherly, is used for control flight following flight plan *(G_{10})*. Suppose N_i is a mean for N_j ($N_i \longmapsto N_j$), then the evidence value of N_j follows with the one in N_i following:

$$Sat(N_j) = Sat(N_i) \quad (9)$$

$$Den(N_j) = Den(N_i) \quad (10)$$

$$X(N_j) = X(N_i) \quad (11)$$

Later, tasks and resources might be analyzed using decomposition relations with the similar principles with the ones for goal-decomposition.

Moreover, the satisfaction/denial of goals, tasks, or resources can effect the other constructs in the model. To capture this situation, the GR framework introduces contribution relations which are adapted from probabilistic causation [31] in the philosophy community. For instance, defining flight plans in ad-hoc (G_{17}) will create difficulty in identifying a resolution action (T_8) in case there is a deviation between a flight and its flight plan, but the denial of G_{17} does not effect either SAT or DEN of T_8 . This setting is modeled as $G_{17} \overset{-0.01S}{\longmapsto} T_8$. It infers the satisfaction of G_{17} will increase the opposite evidence (DEN) of T_8 until 0.01. Essentially, the GR framework distinguishes contribution relations into 4 types: positive-satisfaction (+wS), positive-denial (+wD), negative-satisfaction (-wS), and negative-denial (-wD), where w represents the extent ($[0 \ldots 1]$) of the effect of source-node to the target-node. "+wS" ("+wD") is used when the satisfaction (denial) of source-node will increase the satisfaction (denial) evidence of the target-node. Conversely, "-wS" ("-wD") models that the satisfaction (denial) of source-node will increase the denial (satisfaction) evidence of the target-node. The relation like $N_i \overset{+w}{\longmapsto} N_j$ is only a shortcut to represent $N_i \overset{+wS}{\longmapsto} N_j$ and $N_i \overset{+wD}{\longmapsto} N_j$. The evidence value of a target-node, as results of contribution relations, is calculated following (12)-(13).

$$q_{Sat(N_j)} = \lceil Sat_0(N_j) + \sum_{i \in src_S} |+w_{ij}| \times Sat(N_i) + \sum_{i \in src_D} |\text{-}w_{ij}| \times Den(N_i) \rceil \quad (12)$$

$$q_{Den(N_j)} = \lceil Den_0(N_j) + \sum_{i \in src_D} |+w_{ij}| \times Den(N_i) + \sum_{i \in src_S} |\text{-}w_{ij}| \times Sat(N_i) \rceil \quad (13)$$

Essentially, (12) (or (13)) calculates the total SAT (or DEN) value of N_j , denoted as $q_{Sat(N_j)}$ (or $q_{Den(N_j)}$), from all contribution relations to N_j . The calculation of $q_{Sat(N_j)}$ (12) is done on the basis of the summation: the initial SAT of N_j ($Sat_0(N_j)$), all satisfaction evidence delivered from all nodes (src_S) connected with positive-satisfaction (+wS) contribution relations, and all denial evidence from all nodes (src_D) connected with negative-denial (-wD) ones. The similar principle is also applied for the calculation of $q_{Den(N_j)}$ (13) Since $q_{Sat(N_j)}$ and $q_{Den(N_j)}$ represent the evidence value of a node from contribution relations, then their value must satisfy the rule (1) but no need to fulfill the rule (2). Normalization methods (14)-(16) are introduced to obtain SAT, DEN , and X of N_j . Essentially, $Sat(N_j)$ and $Den(N_j)$ is defined as the value of $q_{Sat(N_j)}$ or $q_{Den(N)}$ respectively, when their summation is at most 1. In the other case (i.e., the summation is greater than 1), one may assume there is a conflict of evidence because the same evidence is counted as satisfaction evidence and denial

evidence. In this framework, the conflicted evidence is counted as $X(N_j)$ because we can decide either it is satisfaction or denial evidence (i.e., $(q_{Sat(N_j)} + q_{Den(N_j)} - 1)$). Consequently, $Sat(N_j)$ is obtained by subtracting $q_{Sat(N_j)}$ with the value of conflicted evidence ($X(N_j)$).

$$Sat(N) = \begin{cases} q_{Sat(N)}, & \text{if } q_{Sat(N)} + q_{Den(N)} \leq 1; \\ 1 - q_{Den(N)}, & \text{otherwise.} \end{cases} \quad (14)$$

$$Den(N) = \begin{cases} q_{Den(N)}, & \text{if } q_{Sat(N)} + q_{Den(N)} \leq 1; \\ 1 - q_{Sat(N)}, & \text{otherwise.} \end{cases} \quad (15)$$

$$X(N) = \begin{cases} 1 - (q_{Sat(N)} + q_{Den(N)}), & \text{if } q_{Sat(N)} + q_{Den(N)} \leq 1; \\ (q_{Sat(N)} + q_{Den(N)} - 1), & \text{otherwise.} \end{cases} \quad (16)$$

While doing this analysis, analysts should be aware with the distinction between *probabilistic causation* and *probabilistic correlation*. *Probabilistic causation* represents the occurrence of source node will effect the probability of the source node, such as smoking is probable caused of lung cancer. Conversely, *probabilistic correlation*, just, models the correlation between the occurrence of source-node and target-node. The present of a yellow stain in a hand increases the chance of lung cancer of hand owner, but it is ridiculous to state that a yellow stain contributes to the increase of having lung cancer. It is because smoking is the one that increases the likelihood of having a yellow stain and a lung cancer. To avoid this fallacies, analysts should verify this analysis whether it fulfills the characteristics that are define in [21]. After analysts do all the modeling analysis, ones know how the stakeholders' interests will be fulfilled. Analysts may associate leaf-nodes with their costs, so that the reasoner will elicit the "most" cost-effective solution to achieve the stakeholders' interests as proposed in [33]. The following modeling (i.e., event modeling) is meant to introduce any uncertain events that may effect the fulfillment of stakeholders' interests.

3.2 Event Modeling

In the GR framework, we adopt the WordNet[4] definition for event:

- something that happens at a given place and time;
- a special set of circumstances;
- a phenomenon located at a single point in space-time;
- a consequence; i.e., a phenomenon that follows and is caused by some previous phenomena.

The notion of *threat* [30] in computer security and *hazardous* condition in reliability engineering [27] are slightly different with the notion of *event* in GR framework. Those concepts are only defined as a potential circumstance that could cause harm or loss and not specifying the notion of likelihood.

The GR framework characterizes events with two properties: likelihood and severity adopting from Probabilistic Risk Assessment (PRA) [5] Likelihood is modeled as a

[4]http://wordnet.princeton.edu/

property of an event which is calculated from the evidence values, whereas severity is denoted as the sign (w) (negative/positive) of an impact relation. Essentially, an impact relation has the same intuition with contribution relations, but it uses the likelihood (λ) instead of SAT/DEN. By modeling severity as the sign of impact relation, it allows us to model situations where an event impacts on more than a single construct with different severity. For instance, in Fig. 4 the event runway incursion *(E_8)* increases the SAT of the event of having collision between aircraft *(E_3)* and collision between aircraft and others *(E_4)* with different severity. Adopting from [9], we define a risk as an event with a negative effect, alternatively an opportunity when it produces positive effects, and consequently w is defined as $[-1 \dots 1]$. $w = [-1 \dots 0)$ represents that the occurrence of source-node is a risk for the occurrence of target-node, $w = (0 \dots 1]$ uses to model an opportunity, and $w = 0$ represents the event does not deliver any impacts to the target-node. This flexibility allows an analyst to model an event which acts as a risk and an opportunity at the same time. It is because an analyst should realize that it is not convenient to eliminate totally the risk, since the event introduces also advantages, so it would better to mitigate its negative effects until an acceptable level. In this case study (Fig. 4), we annotate the events with SAT which is obtained from EUROCONTROL reports [14, 37][5]. Essentially, those reports are resulted from the statistical analysis of its historical data of aviation in European airspace. Therefore, we assume the DEN can be defiend as $1 -$ SAT (i.e., there is no ignorance X).

In this framework, an event is modeled as a *states* which is held on a particular *time*. For stating the event mid-air collision *(E_1)*, we mean that:

$$E_1 = (S_1, t_x) \text{ where } S_1 = \text{mid-air collision}, \ t_x \in \text{time-instance}$$

This form distinguishes an event from a goal as a state-of-affair that an actor intends to achieve. Suppose an actor has the goal having mid-air collision (S_1). When S_1 is held at time t_1 and t_2, so we may argue they are the same goal and loosing the fact that it happens twice. Differently, if we model S_1 as an event then the occurrences of the same state (S_1) at different time are counted as different event. The identification of events (called top-events) can be realized using different approaches, such as obstacle analysis [40], anti-goal [39], hazard analysis [24], misuse case [35], abuse case [29], or taxonomy-base risk identification [8]. After identifying top-events, analysts analyze them using the similar steps as in the goal layer (i.e., decomposition analysis and contribution analysis). Events are decomposed (i.e., AND/OR) into sub-events until reach leaf-events. A proper leaf-event may be determined by:

- an event can not refined into distinguished/disjoin sub-events;
- its sub-events is difficult to be assessed (e.g., their likelihood are too small, lack of historical data)

For instance, the event mid-air collision *(E_2)* is AND-decomposed into imminent collision-IC *(E_6)* and ineffective collision avoidance *(E_7)*. In event layer, an event AND-decomposition must represent the order of the occurrences of sub-events to result the up-level event. It is because if the E_7 occurs before E_6 then it will not result in

[5]the annotation also represents the number of an event occurrence from a million hour of flight

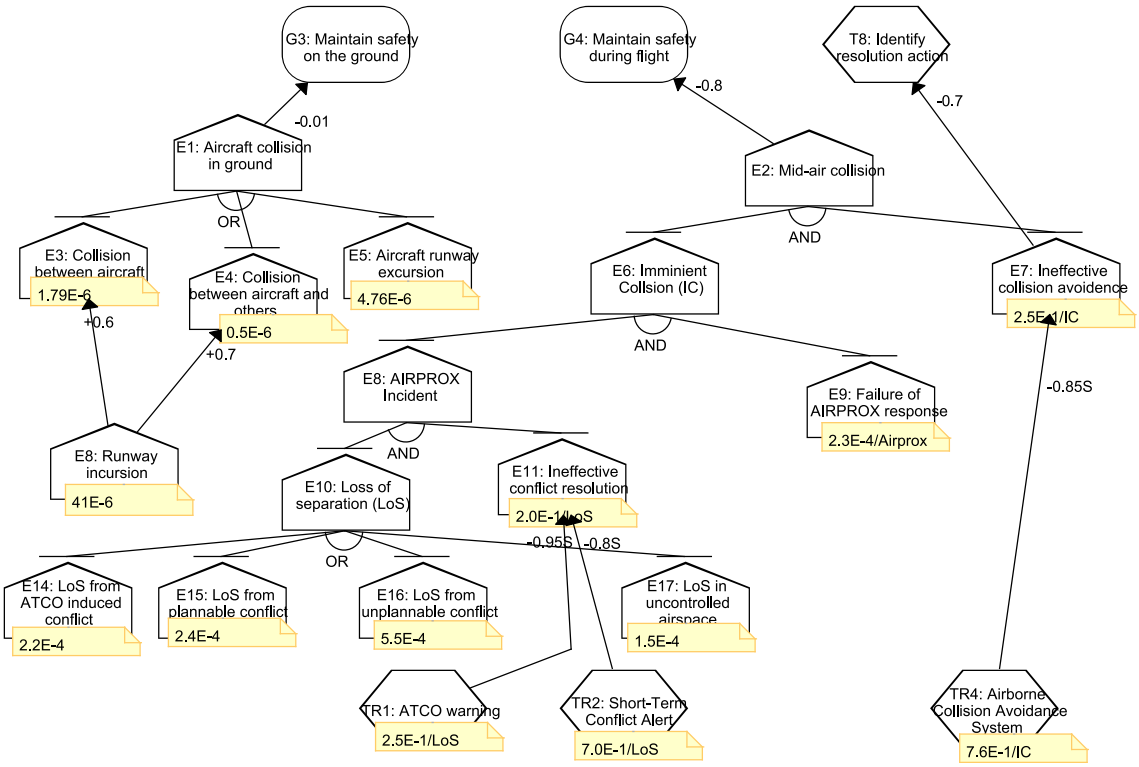


Figure 4: Event and Treatment Layers of ATM Scenario

E_2 . For this purpose, graphically we assume the most-left sub-event is the first sub-event that must be occurred, and the most-right sub-event is the last one. Suppose we define E_6 as (S_6, t_6) and E_7 as (S_7, t_7), we should specify that t_7 occurs after S_6 is held (i.e., $t_7 \gg S_7$). Consequently, the SAT of E_7 is defined assuming the event E_6 has occurred; $Sat(7) = 2.5E-1$ /Imminent Collision. In OR-decomposition, the sub-events should be disjoint events and no need to specify their order. The similar mathematical models (3)-(8) is used to calculate the evidence values of an up-level event.

Afterwards, analysts model interdependency (i.e., probabilistic causation) from an event to other nodes (i.e., goals, tasks, resources, and events) using impact relations, such as $E_1 \overset{-0.01}{\longmapsto} G_3$ or $E_8 \overset{+0.6}{\longmapsto} E_3$ (Fig. 4). Essentially, an impact relation adds the evidence values (i.e., SAT and DEN) of the target-node according to the likelihood of the event and the severity of the relation. To calculate the likelihood of an event ($\lambda(E)$), we must decide whether an event is categorized as a risk or a opportunity. The following formula can be used to decide the categorization of an event:

$$\sum_{R_i \in \text{impact-rels from } E} w_i > 0 \ \textit{iff}\ E \ \text{ is an opportunity}$$

R_i is an impact relation that is originated from E. The event may have several impact relations, if the total of severity (w_i) of all impact relations (R_i) is greater than 0 then E is categorized as an opportunity, otherwise it is a risk.

As mentioned before, the likelihood of an event ($\lambda(E)$) is calculated on the basis of the evidence values of an event which is defined as follow:

$$\lambda(E) = \begin{cases} Sat(E), & \text{if } E \text{ is an opportunity;} \\ Sat(E) + X(E)^6, & \text{otherwise.} \end{cases} \quad (17)$$

Essentially, $\lambda(E)$ calculation (17) is calculated assuming the worst condition by counting the ignorance ($X(E)$) as the supporting evidence for an event which is judged as a risk. The similar mathematical models, with the ones for contribution relations, are defined in (18)-(19) to calculate the evidence values from impact relations.

$$q_{Sat(N_j)} = \lceil Sat_0(N_j) + \sum_{i \in src} |+ w_{ij}| \times \lambda(N_i) \rceil \quad (18)$$

$$q_{Den(N_j)} = \lceil Den_0(N_j) + \sum_{i \in src} |\text{-}w_{ij}| \times \lambda(N_i) \rceil \quad (19)$$

To obtain SAT,DEN, and X, q values must be normalized using the normalization methods (14)-(16). By means of this modeling, analysts understand how events are developed until they impact to the goal layer. The model allows analysts to assess their impacts to their goal event, and, possibly, to define the criticality of the events.

3.3 Treatment Modeling

Once the goal and event layers have been analyzed, analysts continue to identify and analyze the countermeasures to be adopted in order to mitigate risks in the GR model. In [2], we have defined the guidelines to identify the treatments from the structure of an event-tree in the event layer. Essentially, treatments/countermeasures are tasks that are meant to mitigate the risk. Therefore, they might be analyzed using (AND/OR) decomposition and contribution relations, the same as the ones in the goal layer. Treatments operates in two different ways: reducing the *likelihood* or reducing the *severity* of an event. To reduce the likelihood, a countermeasure is modeled using a contribution relation which introduces denial evidence to an event. For instance, the treatment Short Term Conflict Alert *(TR_2)* adds denial evidence for the event ineffective conflict resolution *(E_{11})*, applying rules on (12)-(13).

To reduce the impact, we introduce the *alleviation* relation as denoted in Fig. 2. This relation intends to reduce the severity (w) of an impact relation; in Fig. 2 for example, the relation between the treatment STCA warning system *(TR)* to the impact relation between the event minima separation infringement *(E)* and the goal maintain safe separation *(G)*. This relation is not intended to reduce the likelihood of the event E, but rather to reduce the severity of the event E (i.e., $TR \overset{y=-0.8}{\longmapsto} [E \overset{w=-2.3E-6}{\longmapsto} G]$)

[6]Based on the rule (2) $Sat(E)$ +X(E)=1- $Den(E)$

by reducing the w into a smaller value, following (20).

$$w = \lfloor w_0 - \sum_{R_i \in \text{alleviate-rels of}[E \overset{w_0}{\longmapsto} N]} y_i \times Sat(TR_i) \rfloor \quad (20)$$

w_0 is the initial severity of a impact relation ($E \overset{w_0}{\longmapsto} N$). For all R_i, which are alleviation relations for the impact relation. We calculate the final severity of impact relation (w) with considering the satisfaction evidence of treatments (TR_i) and the weight (y_i)[7] of the alleviation relation. Moreover, the severity (w) of an impact relation is never be below 0, and $w = 0$ indicates there is no impact between the event and the goal. In our model, we also allow for relations between the treatment layer and the goal layer. This is useful to model situations where a countermeasure is adopted to mitigate a risk and has also a contribution (especially negative) to the goal layer. Finally, we have already completed our proposal introducing a quantitative model in the Tropos GR framework using the ATM scenario. The similar analysis, as we have proposed in [3], can be done in this model. Such analysis helps analysts to elicit requirements to be realized in the system-to-be. The requirements has already analyzed the risk that might be introduced and necessary countermeasures have already been incorporated. It will minimize the likelihood of requirement revision due to an unacceptable risk.

4 Related Work

Related work lies on three major areas: requirement engineering, secure and dependable engineering, and risk analysis. In **requirement engineering**, Dardenne et al. [10] propose KAOS, a goal-oriented requirements engineering methodology aiming at modeling not only *what* and *how* aspect of requirements but also *why*, *who*, and *when*. KAOS introduces also the concept of *obstacles* [40] and *anti-goal* [39] 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. Mayer et al. [28] extend the *i** conceptual framework [44] 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.

In the area of **secure and dependable system**, the most used frameworks are the classical ones, namely Fault Tree Analysis (FTA) [38], Failure Modes, Effects, and Criticality Analysis (FMECA) [12]. In security engineering, approaches like *attack tree* and *threat tree* [32, 20] are similar to the FTA. However, the most relevant work for our purpose is Defect Detection and Prevention (DDP) by Feather et al. [16] has been developed and applied in Jet Propulsion Lab of NASA. DDP consists of a three layers model: Objectives, Risks, and Mitigation. Each objective has a *weight* to represent its importance, each risk has a *likelihood* of occurrence, while mitigation has a *cost* for its accomplishment (mainly resource consumption). Severity of a risk can be

[7] $y = [-1 \ldots 0)$

represented by an impact relation between the objective and the risk. Moreover, a DDP model specifies how to compute the level of objectives achievement and the cost of mitigations. This calculation allows one to evaluate the impact of taken countermeasures and then support the decision making process.

In the area of **risk analysis**, uncertain events (i.e., threats and failures) are quantified with two attributes: likelihood and severity. Probabilistic Risk Analysis (PRA) [5] is widely used for quantitatively risk assessment, while approaches like FMECA [12] quantify risk into qualitative values: frequent, reasonable probable, occasional, remote, and extremely unlikely. Basically, events are prioritized using the notion of "expectancy loss" which is a multiplication between the likelihood of events and its severity. Approaches like Multi-Attribute Risk Assessment [7] 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. Finally, CORAS [11] is aiming at developing a framework for risk analysis of security critical systems. The CORAS risk management consists of the following steps: context identification, risk identification, risk analysis, risk evaluation, and risk treatment.

5 Concluding Remarks

In this paper, we have presented a framework to model risk in an organizational-setting using quantitative data. This framework allows analysts to use objective data (e.g., historic-statistical data) or subjective data (e.g., expert judgment). In case of using objective data, the analysts may eliminate ignorance ($X = 0$) in their calculation. We have adopted our qualitative reasoning algorithms [3] to analyze risk during the process of evaluation and selection of alternatives using quantitative data. Due to the limitation of the pages, we will publish our complete framework and its application in ATM scenario in terms of technical report.

Our approach has some limitations that we would like to overcome in our future work. Particularly, the fact that we only model the *time* aspect of event in terms of the order of event occurrences. We have realized that there could be the case where an event must occur between a particular interval time. Such as, the event near-CFIT (Controlled Flight Into Terrain) of a flight occurs when there is the event MSAW (Minimum Safe Altitude Warning) and followed by the event change flight heading occurs within 20 seconds. Besides that, this framework has not model interrelationships (i.e., delegation, trust) among actors/stakeholders which are essential for a socio-technical system. Based on our current work in [4], we intend to adapt that framework in to our qualitative analysis. Finally, the existence tool is really important to help an analyst realizing this proposed framework.

References

[1] A. I. Anton. Goal-Based Requirements Analysis. In *Proceedings of the 2nd IEEE International Conference on Requirements Engineering (ICRE'96)*, page 136, Washington, DC, USA, 1996. IEEE Computer Society Press.

[2] Y. Asnar and P. Giorgini. Modelling Risk and Identifying Countermeasures in Organizations. In *Proceedings of 1st International Workshop on Critical Information Infrastructures Security (CRITIS '06)*, volume 4347 of *Lecture Notes in Computer Science*, pages 55–66. Springer-Verlag, 2006.

[3] Y. Asnar and P. Giorgini. Risk Analysis as part of the Requirements Engineering Process. Technical Report DIT-07-014, DIT - University of Trento, March 2007.

[4] Y. Asnar, P. Giorgini, F. Massacci, and N. Zannone. From Trust to Dependability through Risk Analysis. In *Proceedings of the Second International Conference on Availability, Reliability and Security*. IEEE Press, 2007.

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

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

[7] S. Butler and P. Fischbeck. Multi-Attribute Risk Assessment. Technical Report CMU-CS-01-169, Carnegie Mellon University, December 2001.

[8] M. J. Carr, S. L. Konda, I. Monarch, F. C. Ulrich, and C. F. Walker. Taxonomy-Based Risk Identification. Technical Report CMU/SEI-93-TR-6, Software Engineering Institute, Carnegie Mellon University, June 1993.

[9] COSO. *Enterprise Risk Management - Integrated Framework*. Committee of Sponsoring Organizations of the Treadway Commission, September 2004.

[10] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-Directed Requirements Acquisition. *Science of Computer Programming*, 20:3–50, 1993.

[11] F. den Braber, T. Dimitrakos, B. A. Gran, M. S. Lund, K. Stølen, and J. Ø. Aagedal. The CORAS Methodology: Model-Based Risk Assessment using UML and UP. In *UML and the Unified Process*, pages 332–357. Idea Group Publishing, 2003.

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

[13] D. Dubois and H. Prade. On The Combination of Evidence in Various Mathematical Frameworks. *Reliability Data Collection and Analysis*, 1992.

[14] EEC - Eurocontrol Experimental Centre. Main Report for the 2005/2012 Integrated Risk Picture for Air Traffic Management in Europe. Technical Report EEC Note No. 05/06, EUROCONTROL, April 2006.

[15] EUROCONTROL. Operational Concept Document: The Vision. Technical Report FC0.ET1.ST07.DEL01, EUROCONTROL, January 2004.

[16] M. S. Feather. Towards a Unified Approach to the Representation of, and Reasoning with, Probabilistic Risk Information about Software and its System Interface. In *Proceedings of the 15th IEEE International Symposium on Software Software Reliability Engineering*, pages 391–402. IEEE Computer Society Press, November 2004.

[17] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Requirements Engineering for Trust Management: Model, Methodology, and Reasoning. *International Journal of Information Security*, 5(4):257–274, 2006.

[18] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani. Formal Reasoning Techniques for Goal Models. *Journal of Data Semantics*, 1(1):1–20, October 2003.

[19] P. Giorgini, J. Mylopoulos, and R. Sebastiani. Goal-Oriented Requirements Analysis and Reasoning in the Tropos Methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, March 2005.

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

[21] C. Hitchcock. Probabilistic Causation. http://plato.stanford.edu/, Februart 2002.

[22] ICAO. *Procedures for Air Navigation Services - Air Traffic Management*. International Civil Aviation Organization, 14 edition, November 2001.

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

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

[25] L. Liu, E. S. K. Yu, and J. Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting. In *Proceedings of the 11th IEEE International Requirements Engineering Conference*, pages 151–161, 2003.

[26] Y.-Q. Liu, Y.-W. Chen, F. Gao, and G.-P. Jiang. Risk Evaluation Using Evidence Reasoning Theory. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, volume 5, pages 2855–2860, 2005.

[27] R. R. Lutz and R. M. Woodhouse. Requirements Analysis Using Forward and Backward Search. *Annals Software Engineering*, 3:459–475, 1997.

[28] N. Mayer, A. Rifaut, and 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.

[29] J. McDermott and C. Fox. Using Abuse Case Models for Security Requirements Analysis. In *Proceedings of 15th Annual Computer Security Applications Conference*, pages 55–64, Phoenix, AZ, USA, 1999.

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

[31] W. C. Salomon. Probabilistic Causality. In E. Sosa and M. Tooley, editors, *Causation*, Oxford Readings in Philosophy. Oxford Press, 1993.

[32] B. Schneier. Attack Trees: Modeling Security Threats. *Dr. Dobbs Journal*, 12(24):21–29, 1999.

[33] R. Sebastiani, P. Giorgini, and J. Mylopoulos. Simple and Minimum-Cost Satisfiability for Goal Models. In *Proceedings of the 16th Conference On Advanced Information Systems Engineering*, volume 3084 of *Lecture Notes in Computer Science*, pages 20–33. Springer-Verlag Heidelberg, June 2004.

[34] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

[35] G. Sindre and A. L. Opdahl. Eliciting Security Requirements With Misuse Cases. *Requirements Engineering Journal*, 10(1):34–44, Jan. 2005.

[36] I. Sommerville. *Software Engineering*. Addison Wesley, 7th edition, May 2004.

[37] SRC - Safety Regulation Commission. Annual Safety Report 2006. Technical report, EUROCONTROL, November 2006.

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

[39] A. van Lamsweerde, S. Brohez, R. D. Landtsheer, and D. Janssens. From System Goals to Intruder Anti-Goals: Attack Generation and Resolution for Security Requirements Engineering. In *Proceedings of the 2nd International Workshop on Requirements for High Assurance Systems*, 2003.

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

[41] P. P. Wakker. Dempster Belief Functions are based on the Principle of Complete Ignorance. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8(3):271–284, 2000.

[42] R. Yager. On the Dempster-Shafer Framework and New Combination Rules. *Information Sciences*, 41(2):93–137, 1987.

[43] R. R. Young. *The Requirements Engineering Handbook*. Artech House, 2004.

[44] E. Yu. *Modelling Strategic Relationships for Process Engineering*. PhD thesis, University of Toronto, Department of Computer Science, 1995.