From Trust to Dependability through Risk Analysis

Yudistira Asnar
University of Trento
yudis@dit.unitn.it

Paolo Giorgini
University of Trento
giorgini@dit.unitn.it

Fabio Massacci
University of Trento
massacci@dit.unitn.it

Nicola Zannone
University of Trento
zannone@dit.unitn.it

Abstract

The importance of critical systems has been widely recognized and several efforts are devoted to integrate dependability requirements in their development process. Such efforts result in a number of models, frameworks, and methodologies that have been proposed to model and assess the dependability of critical systems. Among them, risk analysis considers the likelihood and severity of failures for evaluating the risk affecting the system.

In our previous work, we introduced the Tropos Goal-Risk framework, a formal framework for modeling, assessing, and treating risks on the basis of the likelihood and severity of failures. In this paper, we refine this framework introducing the notion of trust for assessing risks on the basis of the organizational setting of the system. The assessment process is also enhanced to analyze risks along trust relations among actors. To make the discussion more concrete, we illustrate the framework with a case study on partial airspace delegation in Air Traffic Management system.

1. Introduction

Critical systems [25] are ubiquitous in today's interconnected society. For instance, failures in *safety-critical* systems result in life loss, or damage to the environment (e.g., nuclear plant management system); failures in *mission-critical* systems result in failure of goal-directed activities (e.g., spacecraft navigation system); and failures in *business-critical* systems result in economic losses (e.g., bank accounting system). In this scenario, dependability (i.e., availability, safety, reliability, maintainability, integrity) becomes a strong requirement for critical systems [3].

To deploy dependable systems, designers need to detect and remove errors and limit damage caused by failures. Several frameworks have been proposed to model and assess the dependability of critical systems [5, 17, 23]. Most of them analyze all possible failures to deploy systems able to anticipate them even when they are very unlikely or insignificant. In this case, one can argue that the design is not cost-effective and decide to not invest on it.

Risk analysis has been proposed as a solution for prioritizing the strategies to mitigate failures by analyzing their likelihood and effects. This approach allows designers to adopt countermeasures only for the most critical failures. For instance, Fault Tree Analysis (FTA) [26] and Probabilistic Risk Assessment (PRA) [4] analyze failures on the basis of their likelihood and impacts and assess the dependability of the system in terms of its risks. However, these frameworks focus on the system-to-be and do not analyze the organizational setting in which the system itself will operate.

In this work, we propose a refined framework for assessing risk at organizational level over what has been proposed in [1]. The first contribution is the introduction of social relations that makes possible to compute the risk by considering the contributions of different actors. An actor of the system may not have the capabilities to meet his responsibilities by himself, and so he depends on other actors for it. These social relations significantly affect the dependability of high-reliable organizations [8]. The second contribution is the introduction of the notion of trust to extend the risk assessment process. The assignment of responsibilities is typically driven by the level of trust towards other actors [10, 24]. Trust is a *subjective probability* that defines the expectation of an actor about profitable behavior of another actor [10]. A low level of trust increases the risk perceived by the depender about the achievement of his objectives. The new constructs have been formalized so that the risk of the system can be formally analyzed through a tool-supported process. Using the framework proposed in this paper, an actor can assess the risk in delegating the fulfillment of his objectives and decide whether or not the risk is acceptable.

The remainder of the paper is structured as follows. Next, we introduce an Air Traffic Management system used as a running example throughout the paper. In Section 3, we provide a brief description of the Goal-Risk modeling framework and describe the basic concepts that we use for assessing risk in organizational settings. In Section 4, we extend the framework by introducing the notion of trust. In Section 5, we explain how to assess risk based on trust relations. Finally, we discuss related works in Section 6 and conclude in Section 7.

2. A Safety-Critical System

This section introduces the Air Traffic Management (ATM) case study [7] that has been studied in the SERENITY Project[1] for the validation of Security & Dependability patterns. An ATM system is managed by an authorized body, called Air Traffic Control Center (ACC) that provides air traffic control (ATC) services in a particular airspace. ATC services are comprised of controlling aircraft, managing airspace, managing flight data of controlled aircraft, and providing information on air traffic situation.

Suppose that there are two adjacent ACCs (e.g., ACC-A and ACC-B) as depicted in Fig. 1. Each ACC divides its airspace into several adjacent volumes, called *sectors*. For instance, ACC-A divides its airspace into 5 sectors (e.g., 1-A, 2-A, 3-A, 4-A, 5-A) and ACC-B in 2 sectors (e.g., 1-B, 2-B). Each sector is managed by a team, consisting of an Executive Controller (EC) (e.g., Edison is the EC of sector 1-A), and a Planning Controller (PC) (e.g., Paul is the PC of sector 1-A). Each team is responsible for the safety of overflight aircraft in its sector. To ease communications, several adjacent sectors in an ACC are supervised by a Supervisor (SU). In our example, Sector 1-A, 2-A, and 3-A are supervised by Scott, while Susan supervises sector 4-A and 5-A, and Spencer supervises sector 1-B and 2-B.

One day in a summer holiday period, Paul receives a flight bulletin that indicates an air traffic increase in the next 6 hours. Based on the planner analysis, the air traffic will be beyond the threshold that Edison can safely handle. Therefore, Scott must take some precautions to handle this situation. In particular, he has two possibilities:

- Dividing the airspace into smaller sectors. In this case, controllers cover smaller areas, but the supervisor have to supervise a greater number of sectors.
- Delegate part of the airspace to an adjacent supervisor. This can be in either the same ACC or a different ACC.

To apply *airspace division*, Scott must have available resources: a controlling team and a pair of controller workstation, called *Controller Working Position* (CWP), for the team. Unfortunately, in the summer holiday all team and CWPs are occupied to manage existing sectors. Therefore, the only alternative to handle the increase without applying any restrictions to incoming traffic, is *partial airspace delegation*. Based on the Paul analysis, Scott can delegate the management of the increase airspace (indicated in Fig. 1) either to Susan or Spencer. Before proceeding, Scott must be sure that the target supervisor (e.g., Susan or Spencer) has infrastructures (e.g., radar and radio coverage) to provide ATC services in the increased airspace and define a delegation schema to rule the partial airspace delegation.

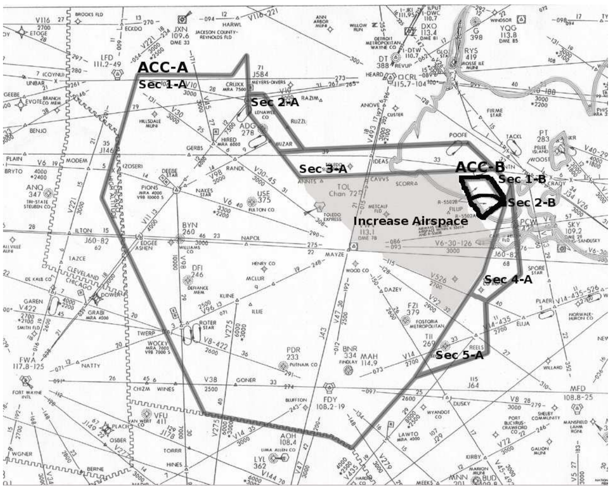

Figure 1. Airspace Division between ACC-A and ACC-B[3]

Actually, Scott has different expectation from the different supervisors due to the personal closeness, the easiness in communication, and air traffic similarities. For instance, Scott and Susan work in the same ACC so that they should not have problems in the coordination of the increased airspace during partial airspace delegation. Conversely, the air traffic in sector 1-B has many similarities with the one in the increased airspace. Therefore, from Scott's perspective, Spencer can handle the traffic in the increased airspace more efficiently.

To decide to whom *increase airspace* should be delegated, Scott needs to assess the risks of each alternative. To support the management of critical systems, we propose a framework for assessing risks using trust relations among actors as evidence besides the capabilities of service providers.

3. Tropos Goal-Risk Framework

The Tropos Goal Risk Model (GR-Model) [2] represents requirements models as graphs $\langle \mathcal{G}, \mathcal{R} \rangle$, where $\mathcal{G}$ are nodes and $\mathcal{R}$ are relations. $\mathcal{G}$ is comprised of three constructs: *goal*, *task*, and *event*. Goals (depicted as ovals) are strategic interests that actors intend to achieve. Events (depicted as pentagons) are uncertain circumstance out of the control of

[1] EU-IST-IP 6th Framework Programme - SERENITY 27587 - http://www.serenity-project.org

[3] The figure is a modification version from http://tol.natca.org

actors that can have an impact on the achievement of goals. Tasks (depicted as hexagons) are sequences of actions used to achieve goals or to treat the effects of events. TEach of above constructs is characterized by two attributes: SAT and DEN. Such attributes represent the values[4] of evidence that the goal can be satisfied and respectively the evidence that the goal can be denied. Their values are qualitatively represented in the range of $\{$*(F)ull,(P)artial,(N)one*$\}$, with the intended meaning $F > P > N$. $\mathcal{R}$ consists of *AND/OR decomposition* and *contribution* relations. AND/OR decomposition relations are used to refine goals, tasks, and events in order to produce a finer structure. Contribution relations are used to model the impacts of a node over another node. We distinguish 4 types of contribution relations: $+$,$++$,$-$, and $--$. Each type can propagate one type of evidence, either SAT or DEN, or both types of evidence. For instance, the $++$ contribution relation indicates that the relation delivers both SAT and DEN, whereas the $++_S$ contribution relation means the relation only delivers SAT evidence to the target goal.

The GR-Model consists of three conceptual layers of analysis [2] as shown in Fig. 2:

Goal layer analyzes the goals of each actor and identifies which tasks the actor needs to perform to achieve the goals;

Event layer models uncertain events along their effects to the goal layer;

Treatment layer identifies specific tasks (also called treatments) that should be introduced to change the consequences of effect of event layer (i.e., mitigate) towards goal layer.

In this paper we extend the GR-Model to support risk analysis beyond the rationale of single actors. To this intent, we introduce the notion of actor in the GR-Model. The formal definition of GR-Model becomes $\langle(\mathcal{A},\mathcal{G}),\mathcal{R}\rangle$ where $\mathcal{A}$ is a set of actors. The extended GR-Model allows us to compute the evidence of fulfillment of the same goal from the perspective of different actors. For instance, Spencer may have *full* evidence that goal manage sector 1-A with the support of another SU (G_{1b}) will be satisfied, whereas Scott may have only *partial* evidence that G_{1b} will be satisfied.

This extension requires refining the predicates used to represent SAT and DEN values, as follow:

- $FS(A,N)[FD(A,N)]$: actor A has (at least) *full* evidence that node N will be satisfied [denied];
- $PS(A,N)[PD(A,N)]$: actor A has (at least) *partial* evidence that node N will be satisfied [denied];
- $NS(A,N)[ND(A,N)]$: actor A has *none* evidence that node N will be satisfied [denied]

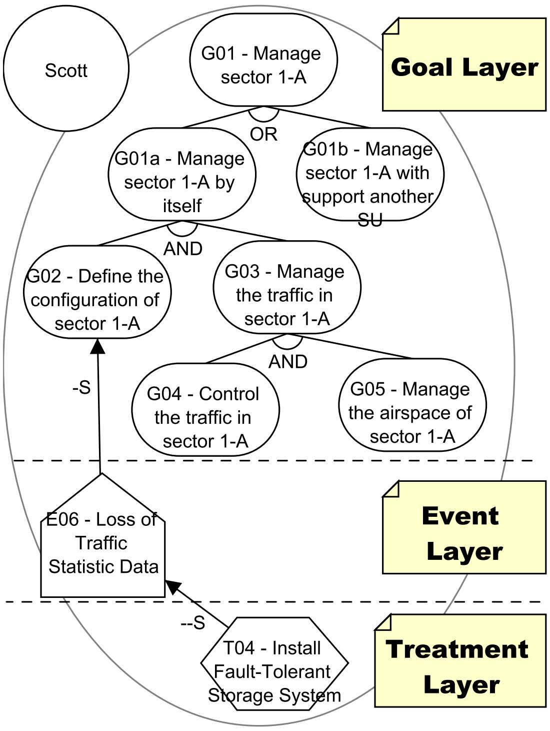
Scott
G01 - Manage sector 1-A
Goal Layer
OR
G01a - Manage sector 1-A by itself
G01b - Manage sector 1-A with support another SU
AND
G02 - Define the configuration of sector 1-A
G03 - Manage the traffic in sector 1-A
AND
-S
G04 - Control the traffic in sector 1-A
G05 - Manage the airspace of sector 1-A
E06 - Loss of Traffic Statistic Data
Event Layer
--S
T04 - Install Fault-Tolerant Storage System
Treatment Layer

Figure 2. Goal-Risk Model of ATM case study

Relations among nodes are represented as $((A_1,N_1),\ldots,(A_1,N_n)) \xrightarrow{r} (A_2,N)$ where r can be a contribution or decomposition relation, $(A_1,N_1),\ldots,(A_1,N_n)$ are called *source nodes*, and (A_2,N) is the *target node* of relation r. All source nodes must belong to the same actor, while the target node can be referred to a different actor. In decomposition relations, source nodes and target nodes must belong to the same actor, while in contribution relations, they might be in the same actor or different ones.

The axioms to propagate SAT and DEN values over traditional Tropos goal models [16] also need to be revised to accommodate the notion of actor. The new axiomatization is presented in Fig. 3. Axioms (1)-(2) formalize the SAT and DEN propagation over goal models: if a node has (at least) *full* evidence of satisfaction (or denial), it also has (at least) *partial* evidence of satisfaction (or denial). Similarly, a node that has (at least) *partial* evidence, also has (at least) *none* evidence.

Axioms (3)-(8) define how SAT and DEN evidence of nodes are calculated on the basis of the evidence of their AND-subparts. In particular, the SAT evidence of a top node follows the lowest SAT evidence of its subparts (Axioms (3)-(5)), whereas the DEN evidence follows the highest DEN values (Axioms (6)-(8)). For instance, in Fig. 2 Scott AND-decomposes goal manage the traffic in sector 1-A (G_3) into subgoals control the traffic in sector 1-A (G_4) and manage the airspace of sector 1-A (G_5). To satisfy

[4] SAT and DEN are independent attributes, and they are different from the one in Probability Theory (i.e., $P'(E) = 1 - P(E)$).

$$
\begin{array}{rll}
\textbf{Node} & \textbf{Invariant Axioms} & \\
N: & FS(A,N) \rightarrow PS(A,N) \rightarrow NS(A,N) & (1)\\
 & FD(A,N) \rightarrow PD(A,N) \rightarrow ND(A,N) & (2)\\
\textbf{Relation} & \textbf{Relation Axioms} & \\
(N_2,N_3) \xmapsto{and} N_1: & FS(A,N_2) \wedge FS(A,N_3) \rightarrow FS(A,N_1) & (3)\\
 & PS(A,N_2) \wedge PS(A,N_3) \rightarrow PS(A,N_1) & (4)\\
 & NS(A,N_2) \wedge NS(A,N_3) \rightarrow NS(A,N_1) & (5)\\
 & FD(A,N_2) \vee FD(A,N_3) \rightarrow FD(A,N_1) & (6)\\
 & PD(A,N_2) \vee PD(A,N_3) \rightarrow PD(A,N_1) & (7)\\
 & ND(A,N_2) \vee ND(A,N_3) \rightarrow ND(A,N_1) & (8)\\
(N_2,N_3) \xmapsto{or} N_1: & FS(A,N_2) \vee FS(A,N_3) \rightarrow FS(A,N_1) & (9)\\
 & PS(A,N_2) \vee PS(A,N_3) \rightarrow PS(A,N_1) & (10)\\
 & NS(A,N_2) \vee NS(A,N_3) \rightarrow NS(A,N_1) & (11)\\
 & FD(A,N_2) \wedge FD(A,N_3) \rightarrow FD(A,N_1) & (12)\\
 & PD(A,N_2) \wedge PD(A,N_3) \rightarrow PD(A,N_1) & (13)\\
 & ND(A,N_2) \wedge ND(A,N_3) \rightarrow ND(A,N_1) & (14)\\
N_2 \xmapsto{x} N_1: & NS(A_1,N_2) \rightarrow NS(A_2,N_1)^5 & (15)\\
 & ND(A_1,N_2) \rightarrow ND(A_2,N_1) & (16)\\
N_2 \xmapsto{++_S} N_1: & FS(A_1,N_2) \rightarrow FS(A_2,N_1) & (17)\\
 & PS(A_1,N_2) \rightarrow PS(A_2,N_1) & (18)\\
N_2 \xmapsto{+_S} N_1: & PS(A_1,N_2) \rightarrow PS(A_2,N_1) & (19)\\
N_2 \xmapsto{--_S} N_1: & FS(A_1,N_2) \rightarrow FD(A_2,N_1) & (20)\\
 & PS(A_1,N_2) \rightarrow PD(A_2,N_1) & (21)\\
N_2 \xmapsto{-_S} N_1: & PS(A_1,N_2) \rightarrow PD(A_2,N_1) & (22)\\
N_2 \xmapsto{++_D} N_1: & FD(A_1,N_2) \rightarrow FD(A_2,N_1) & (23)\\
 & PD(A_1,N_2) \rightarrow PD(A_2,N_1) & (24)\\
N_2 \xmapsto{+_D} N_1: & PD(A_1,N_2) \rightarrow PD(A_2,N_1) & (25)\\
N_2 \xmapsto{--_D} N_1: & FD(A_1,N_2) \rightarrow FS(A_2,N_1) & (26)\\
 & PD(A_1,N_2) \rightarrow PS(A_2,N_1) & (27)\\
N_2 \xmapsto{-_D} N_1: & PD(A_1,N_2) \rightarrow PS(A_2,N_1) & (28)
\end{array}
$$

Figure 3. SAT **and** DEN **Evidence Propagation**

G_4, Scott must fulfill all these subgoals. Axioms for OR-decomposition (Axioms (9)-(14)) behave conversely from the ones for AND-decomposition. For instance, in Fig. 2 Scott intends to satisfy manage sector 1-A (G_1). This goal can be achieved either by fulfilling manage sector 1-A by itself (G_{1a}) or manage sector 1-A with the support of another SU (G_{1b}). It is sufficient that Scott fulfills one of these OR-subgoals to satisfy G_5.

Axioms (15)-(28) cope with contribution relations. These axioms are applied when contribution relations are both in intra-actor (i.e., source node and target node are laid in the same actor) and inter-actor (i.e., source node and target node are laid in different actors). In particular, when the relation is inter-actor, it means that evidence that an actor has on the satisfaction or denial of a goal affect the evidence that another actor has on the satisfaction or denial of his goals. In particular, axioms (15)-(16) state that nodes that do not have any evidence do not deliver evidence on the satisfaction or denial of other nodes. Axioms (17)-(28) propagate SAT or DEN evidence from the source node to the target node according to the type of contribution.

4. Trust in GR Model

An actor might not have all capabilities to fulfill his goals and tasks. Tropos introduces the notion of *dependency* to deal with this issue. In [14], we proposed a conceptual refinement of dependency by introducing the notions of *delegation* and *trust*. Delegation is used to model the transfer of responsibilities from an actor (the *delegator*) to another (the *delegatee*). By delegating the fulfillment of a goal, the delegator becomes vulnerable because, if the delegatee fails to fulfill the assigned responsibilities, the delegator will not be able to achieve his objectives. Thereby, such a situation introduces risks that decrease the dependability of the system. Trust relations are used to model the expectation of an actor (the *trustor*) about the behavior of another actor (the *trustee*) in achieving a goal. Together with the notion of trust, we adopt also the notion of *distrust* [15]. This relation is used to model the belief of an actor about the misbehavior of another actor in achieving a goal.

We intend to assess the risk beyond the perspective of single actors by adopting the notions of delegation (**D**), trust (**T**) and distrust (**S**) in addition to contribution and decomposition. Indeed, trust and distrust relations can be seen as potential evidence for assessing the risks [10]. Trusting another actor implies that the trustor has considerable *subjective probability* that the trustee will fulfill his responsibility towards the trustor. Trust and distrust relations are indicated by ternary predicates *trust-rel* and *distrust-rel*, respectively. The first parameter represents the trustor, the second the trustee, and the last the goal intended to be achieved or the task intended to be executed. We also introduce the notion of *trust level* that allows us to simplify later notation. In particular, we have considered three trust levels: *Trust*, *Distrust*, and *NTrust* (i.e., neither trust nor distrust). The last is necessary since the requirements specification may not define any trust or distrust relation between two specific actors.

Axioms in Fig. 4 are introduced to calculate the transitive closure of trust relations and the corresponding trust level on the basis of trust relations. We assume the following order of trust: $Distrust > Trust > NTrust$. This choice can be regarded as a particular instantiation of the denial-takes-precedence principle [18]. This corresponds to a conservative approach which discredits all trust relations in presence of a distrust relation.

Axioms (29) and (31) propagate trust/distrust relations over AND/OR refinement. The idea is that if an actor be-

[5] $x \in \{++_S, +_S, --_S, -_S, ++_D, +_D, --_D, -_D\}$; A_1 and A_2 might be the same actor or two different actors

(Dis)Trust Relations

$$\textit{trust-rel}(A_1, A_2, G) \wedge \textit{subgoal}(G_1, G) \rightarrow \textit{trust-rel}(A_1, A_2, G_1) \tag{29}$$

$$\textit{trust-rel}(A_1, A_2, G) \wedge \textit{trust-rel}(A_2, A_3, G) \rightarrow \textit{trust-rel}(A_1, A_3, G) \tag{30}$$

$$\textit{distrust-rel}(A_1, A_2, G) \wedge \textit{subgoal}(G_1, G) \rightarrow \textit{distrust-rel}(A_1, A_2, G_1) \tag{31}$$

$$\textit{trust-rel}(A_1, A_2, G) \wedge \textit{distrust-rel}(A_2, A_3, G) \rightarrow \textit{distrust-rel}(A_1, A_3, G) \tag{32}$$

Trust Level

$$\textit{distrust-rel}(A_1, A_2, G) \rightarrow Distrust(A_1, A_2, G) \tag{33}$$

$$\neg\textit{distrust-rel}(A_1, A_2, G) \wedge \textit{trust-rel}(A_1, A_2, G) \rightarrow Trust(A_1, A_2, G) \tag{34}$$

$$\neg\textit{distrust-rel}(A_1, A_2, G) \wedge \neg\textit{trust-rel}(A_1, A_2, G) \rightarrow NTrust(A_1, A_2, G) \tag{35}$$

Figure 4. Assessing Trust Level

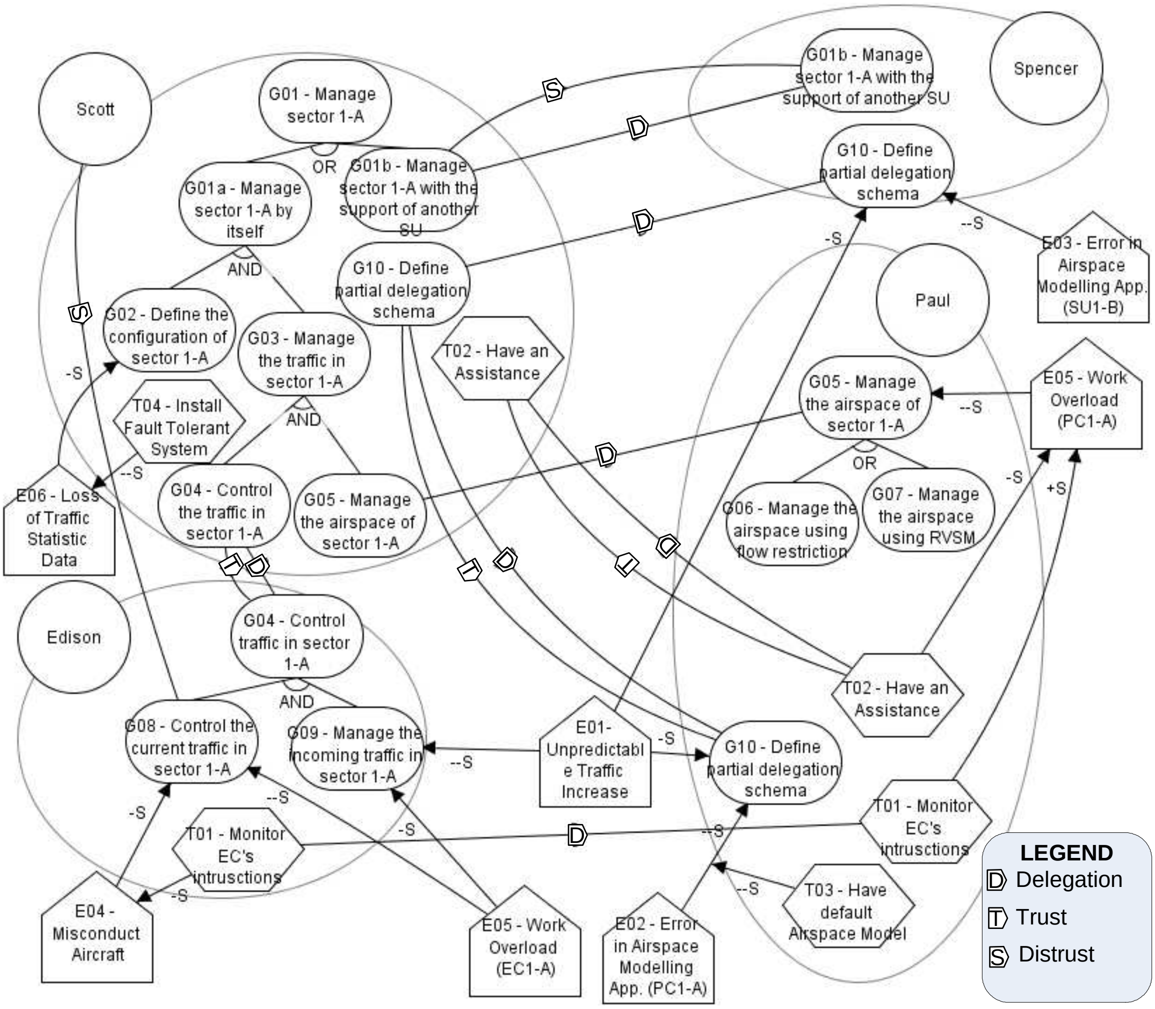
Scott
G01 - Manage sector 1-A
OR
G01a - Manage sector 1-A by itself
G01b - Manage sector 1-A with the support of another SU
G10 - Define partial delegation schema
AND
G02 - Define the configuration of sector 1-A
G03 - Manage the traffic in sector 1-A
T02 - Have an Assistance
T04 - Install Fault Tolerant System
AND
G04 - Control the traffic in sector 1-A
G05 - Manage the airspace of sector 1-A
E06 - Loss of Traffic Statistic Data
-S
--S
S
Spencer
G01b - Manage sector 1-A with the support of another SU
G10 - Define partial delegation schema
E03 - Error in Airspace Modelling App. (SU1-B)
--S
-S
S
D
D
Paul
G05 - Manage the airspace of sector 1-A
OR
G06 - Manage the airspace using flow restriction
G07 - Manage the airspace using RVSM
E05 - Work Overload (PC1-A)
--S
-S
+S
D
T
D
T
D
T
D
T02 - Have an Assistance
G10 - Define partial delegation schema
T01 - Monitor EC's intrusctions
T03 - Have default Airspace Model
--S
-S
--S
Edison
G04 - Control traffic in sector 1-A
AND
G08 - Control the current traffic in sector 1-A
G09 - Manage the incoming traffic in sector 1-A
E01 - Unpredictable Traffic Increase
--S
-S
T01 - Monitor EC's intrusctions
D
-S
--S
-S
-S
E04 - Misconduct Aircraft
E05 - Work Overload (EC1-A)
E02 - Error in Airspace Modelling App. (PC1-A)
LEGEND
D Delegation
T Trust
S Distrust

Figure 5. Extended Goal-Risk Model of ATM case study

lieves that another actor will (not) achieve a goal or execute a task, the first also believes that the latter will (not) fulfill its sub-parts. For instance, in Fig. 5 Scott trusts Edison for achieving goal control the traffic in sector 1-A (G_4). In this setting, Scott also trusts Edison in achieving both goals control the current traffic in sector 1-A (G_8) and manage the incoming traffic in sector 1-A (G_9) which are subgoals (AND-decomposition) of G_4.

Axiom (30) computes the transitive closure of trust relations.[6] It infers indirect relations of trust between two

[6] For the sake of simplicity, we assume that trust is transitive. This choice mainly depends on the qualitative approach adopted in this paper.

$$Trust(A_1, A_2, S) \wedge FS(A_2, S) \rightarrow FS(A_1, S) \tag{36}$$

$$Trust(A_1, A_2, S) \wedge PS(A_2, S) \rightarrow PS(A_1, S) \tag{37}$$

$$Trust(A_1, A_2, S) \wedge NS(A_2, S) \rightarrow NS(A_1, S) \tag{38}$$

$$Trust(A_1, A_2, S) \wedge FD(A_2, S) \rightarrow FD(A_1, S) \tag{39}$$

$$Trust(A_1, A_2, S) \wedge PD(A_2, S) \rightarrow PD(A_1, S) \tag{40}$$

$$Trust(A_1, A_2, S) \wedge ND(A_2, S) \rightarrow ND(A_1, S) \tag{41}$$

$$Distrust(A_1, A_2, S) \rightarrow NS(A_1, S) \tag{42}$$

$$Distrust(A_1, A_2, S) \rightarrow FD(A_1, S) \tag{43}$$

$$NTrust(A_1, A_2, S) \wedge FS(A_2, S) \rightarrow PS(A_1, S) \tag{44}$$

$$NTrust(A_1, A_2, S) \wedge PS(A_2, S) \rightarrow NS(A_1, S) \tag{45}$$

$$NTrust(A_1, A_2, S) \wedge NS(A_2, S) \rightarrow NS(A_1, S) \tag{46}$$

$$NTrust(A_1, A_2, S) \wedge FD(A_2, S) \rightarrow FD(A_1, S) \tag{47}$$

$$NTrust(A_1, A_2, S) \wedge PD(A_2, S) \rightarrow FD(A_1, S) \tag{48}$$

$$NTrust(A_1, A_2, S) \wedge ND(A_2, S) \rightarrow PD(A_1, S) \tag{49}$$

Figure 6. SAT and DEN Evidence Propagation considering Trust Relations

actors. Axiom (32) identifies indirect distrust relations between actors. The idea underlying such an axiom is that, if an actor distrusts another actor, all the actors, who trust the first, distrust the latter.

Trust level is calculated on the basis of the transitive closure of trust and distrust relations drawn by the designer. Axioms (33)-(35) formalize the precedence of the trust level. If there is a distrust relation between two actors, the framework concludes the trust level between them is $Distrust$; if there are only trust relations, the trust level is $Trust$. Finally, if neither trust nor distrust relation has been identified, the trust level is $NTrust$. For instance, in Fig. 5 there are two trust relations between Scott and Edison for goal control the current traffic in sector 1-A(G_8). The first is a direct distrust relation, while the latter is an indirect (i.e., it is inherited from goal G_4 as shown above) trust relation. Since $Distrust$ takes precedence over $Trust$, the trust level between Scott and Edison for achieving G_8 is $Distrust$. These axioms are also used to assess trust level when there are multi-paths of trust between them.

The axioms shown in Fig. 6 extend the formal framework to assess risks by specifying how SAT and DEN evidence are propagated along trust relations. Axioms (36)-(41) cope with situations where the trust level is $Trust$. In this case, the evidence from the trustor viewpoint is the same with the ones of the trustee. For instance, in Fig. 5 Scott trusts Edison to control the traffic in sector 1-A (G_4). If Edison has *full* SAT evidence on G_4 (i.e., $FS(Edison, G_4)$) then Scott has also *full* SAT evidence (i.e., $FS(Scott, G_4)$).

Conversely, if an actor distrusts another actor, the trustor will have *null* SAT evidence and *full* DEN evidence whatever the evidence of the trustee (Axioms (42)-(43)). According such axioms, a delegation in presence of a distrust relation between the delegator and the delegatee is risky for the delegator. For instance, Scott distrusts Spencer to manage sector 1-A (G_{1b}) and Spencer is the one who has evidence about its satisfaction (or denial). From the viewpoint of Scott, goal G_{1b} has *null* evidence of being satisfied (i.e., $NS(Scott, G_{1b})$) and *full* evidence of being denied (i.e., $ND(Scott, G_{1b})$) independently from the evidence in Spencer's viewpoint because Scott does not trust Spencer in fulfilling G_{1b}. Thereby, if Scott must delegate the fulfillment of goal G_{1b} to Spencer, such a delegation is very risky from Scott's perspective. Yet, this may turn out to be the only alternative available at the moment.

Algorithm 1 Risk_Assessment($\langle(\mathcal{A}, \mathcal{G}), \mathcal{R}\rangle$, input_label)

```
Require: goal_model $\langle(\mathcal{A}, \mathcal{G}), \mathcal{R}\rangle$,
    node_matrix $input\_label$ {the initial evidence of each node with cell$_{ij}$
    represents $(Actor_i, Node_j)$ }
 1: TrustBase $\leftarrow calculate\_trust(\langle(\mathcal{A}, \mathcal{G}), \mathcal{R}\rangle)$
 2: current $\leftarrow input\_label$
 3: repeat
 4:     old $\leftarrow$current
 5:     for all $A_i \in \mathcal{A}$ do
 6:         for all $N_j \in \mathcal{G} \wedge requester(N_j) = A_i$ do
 7:             current$_{ij}$ $\leftarrow apply\_rules(i, j, old, \langle(\mathcal{A}, \mathcal{G}), \mathcal{R}\rangle, TrustBase)$
 8:         end for
 9:     end for
10: until {old=current}
```

Finally, axioms (44)-(49) define rules propagating evidence when the trust level is $NTrust$. They reduce SAT evidence and increase DEN evidence.

5. Risk Assessment Algorithm

The assessment process is performed using Algorithm 1. The algorithm calculates SAT and DEN values for each node (*node labels*). The algorithm takes in input a GR-Model $\langle(\mathcal{A}, \mathcal{G}), \mathcal{R}\rangle$ and an $input\_label$, a two-dimension array (i.e., actors, nodes). This array contains initial node labels (e.g., full/partial/null SAT and DEN) from the perspective of each actor. Before assessing risks, the algorithm computes the trust level between actors (line 1) by applying axioms (29)-(35) and stores the result in array $TrustBase$. Then, the algorithm (line 7) applies all the other axioms to collect evidence for all nodes in each actor viewpoint (i.e., N_j is requested by A_i). The process terminates when there is no change between the current labels and the previous ones.

The risk assessment algorithm uses procedure *Apply_Rules* (Algorithm 2) to combine the evidence for the node N_j in actor A_i viewpoint (i.e., (A_i, N_j)). The evidence is computed from all its incoming relations (i.e., decomposition, contribution, and trust relations). Lines 4-5 compute SAT or DEN evidence derived from decomposition/contribution relations. In particular, sat_rules and

More complex trust metrics can be adopted in a quantitative approach.

Algorithm 2 Apply_Rules

```
Require: goal_model $\langle(\mathcal{A},\mathcal{G}),\mathcal{R}\rangle$
 1: for all $R_k \in \mathcal{R} \wedge target(R_k) = (A_i, N_j)$ do
 2:     $(A_{src}, N_{src}) \leftarrow source(R_k)$
 3:     if $type(R_k) \in \{dec, cont\}$ then
 4:         sat$_k$ $\leftarrow sat\_rules(A_i, A_{src}, R_k, N_{src}, old)$
 5:         den$_k$ $\leftarrow den\_rules(A_i, A_{src}, R_k, N_{src}, old)$
 6:     else if $type(R_k) \in \{del\}$ then
 7:         trust $\leftarrow trust\_level(TrustBase, A_{src}, A_i, N_{src})$
 8:         sat-t$_k$ $\leftarrow sat\_rules\_del(A_{src}, A_i, R_k, N_{src}, trust, old)$
 9:         den-t$_k$ $\leftarrow den\_rules\_del(A_{src}, A_i, R_k, N_{src}, trust, old)$
10:     end if
11: end for
12: return { $max(max\_array(sat), max\_array(sat\text{-}t), Old_{ij}.sat)$,
        $max(max\_array(den), min\_array(den\text{-}t), Old_{ij}.den)$ }
```

den_rules use the axioms introduced in Fig. 3 where (A_i, N_j) is the target node, (A_{src}, N_{src}) is source node(s), R_k represents the type of relation, and array old contains the evidence values of the source node(s). The evidence derived in these steps are stored in arrays sat and den, respectively. Lines 7-9 compute the evidence derived from trust relations. When an actor delegates the fulfillment of a goal or the execution of a task to another actor, the algorithm searches the trust level between them in $TrustBase$ (Line 7). Based on such a level, the algorithm calculates the evidence on the basis of trust using the evidence of the delegatee (Lines 8-9). Essentially, sat_rules_del and den_rules_del computes SAT and DEN evidence using the axioms in Fig. 6 and stores them in arrays *sat-t* and *den-t*, respectively.

Line 12 defines how to combine SAT and DEN evidence of nodes. The evidence derived from decomposition/contribution relations are calculated by taking the maximum evidence. The combination of SAT and DEN evidence derived from trust relations is performed differently. An actor (e.g., Scott) might delegate the achievement of a goal (e.g., define partial delegation schema (G_{10})) to different actors (e.g., Spencer and Paul). By assigning the same responsibility to different actors, the delegator is less vulnerable. This reveals that the evidence value of a node should be computed based on all delegation relations by considering the trust levels and the evidence values of each delegatee. Therefore, the SAT evidence are calculated by taking the maximum SAT evidence from all delegatees, and, conversely, the DEN evidence by taking the minimum ones.

The ultimate values of SAT and DEN evidence for node (A, N) are the maximum between the evidence derived from decomposition/contribution and the ones derived from trust relations. The algorithm may compute conflicting SAT and DEN values for a node (e.g., $FS(A, N)$ and $PD(A, N)$). The framework uses a conflict resolution process whose idea is to reduce the value of the SAT and DEN evidence by reducing their values until one of them has *null* evidence. For instance, $FS(A, N) \wedge PD(A, N)$ can become $PS(A, N) \wedge ND(A, N)$. Additional details of conflict resolution are explained in [2].

6. Related Work

Several approaches have been proposed in literature to model risk of critical systems. Mayer et al. [22] extend the *i** modeling framework [27] to analyze risks on security aspects during the development process of IT systems. The framework models the business assets (i.e., goals) of an organization and the assets of its IT system (i.e., architecture, design decisions). Countermeasures to mitigate risks are then selected in such a way that risks do not affect the business assets and the assets of IT system severely. Lee et al. [20] propose a framework for modeling critical systems (especially socio-technical systems) which is supported by a methodology developed by US Department of Defense, called DITSCAP [9]. Both proposals do not assess the level of risk, but only identify its existence.

In the area of risk analysis, there are several models that attempt to quantify uncertain events (i.e., threats, failures) with two attributes: likelihood and severity. Probabilistic Risk Analysis (PRA) [4] is widely used to assess risks quantitatively. Events are prioritized using the notion of "expectancy loss" that is a multiplication between the likelihood of events and its severity. When resources are limited, an analyst can decided to adopt countermeasures for mitigating events on the basis of their priority. Multi-Attribute Risk Assessment [6] improves the risk analysis process by considering multi-attributes. Risk analysis traditionally intends to reduce the risk affecting a system. However, many factors (e.g., reliable, available, safe, etc.) can be critical for a system and each of them has its own risks. This leads analysts to trade-off one attribute to gain lower risk for other attributes. The CORAS [13] methodology combines UML and Unified Process to support a model-based risk assessment. In particular, it proposes an integrated system development and risk management process for security critical systems.

In the area reliability engineering, Defect Detection and Prevention [11, 12] are proposed by Jet Propulsion Lab. (NASA). This framework consists of a three layer model (i.e., objective, risks, and mitigation) which is at the basis of our work. In this model, each objective has a *weight* to represent its importance; a risk has a *likelihood* of occurrence; and a mitigation has a *cost* for accomplishment (namely resource consumption). The DDP model specifies how to compute the level of objectives achievement and the cost of mitigation from a set of given mitigation. This calculation supports designers during the decision making process by evaluating the impact of countermeasures.

Jøsang and Presti [19] explore the relation between risk and trust. This framework defines a notion of trust (*reliability trust* [21]) based on the result of the risk assessment process. The idea is that a trust relation between two actors will be established only if the risk in delegating the fulfillment of a service is acceptable for the delegator. This framework is orthogonal to our approach. Indeed, we use trust as evidence to assess the risk of the system, whereas Jøsang and Presti use risk to assess trust relations among actors.

7. Conclusion

In this paper, we have presented an extension of the Tropos Goal-Risk framework. Particularly, we have proposed an approach to assess risk on the basis of trust relations among actors. The work is still in progress and we are currently working on introducing the notion of permission for assessing dependability of secure systems. Another direction is to extend the framework to support quantitative risk analysis rather than only qualitative analysis.

Acknowledgments

We thank Valentino Meduri, Paola Lanzi, Roberto Bonato, and Carlo Riccucci for many useful discussions on ATM systems. This work has been partially funded by EU Commission, through the SENSORIA and SERENITY projects, by the FIRB program of MIUR under the ASTRO and TOCAI projects, and also by the Provincial Authority of Trentino, through the MOSTRO project.

References

[1] Y. Asnar and P. Giorgini. Modelling risk and identifying countermeasures in organizations. In *Proc. of CRITIS'06*, 2006.

[2] Y. Asnar, P. Giorgini, and J. Mylopoulos. Risk Modelling and Reasoning in Goal Models. Technical Report DIT-06-008, University of Trento, 2006.

[3] A. Avizienis, J.-C. Laprie, B. Randell, and C. E. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *TDSC*, 1(1):11–33, 2004.

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

[5] R. Butler, J. Maddalon, A. Geser, and C. Muñoz. Simulation and Verification I: Formal Analysis of Air Traffic Management Systems: The Case of Conflict Resolution and Recovery. In *Proc. of WSC'03*, pages 906–914. IEEE Press, 2003.

[6] S. A. Butler. Security Attribute and Evaluation Method. PhD thesis, Carnegie Mellon University, 2003.

[7] S. Campadello, L. Compagna, D. Gidoin, P. Giorgini, S. Holtmanns, J. Latanicki, V. Meduri, J.-C. Pazzaglia, M. Seguran, R. Thomas, and N. Zannone. S&D Requirements Specification. Research report A7.D2.1, SERENITY consortium, 2006.

[8] S. Cox, B. Jones, and D. Collinson. Trust Relations in High-Reliability Organizations. *Risk Analysis*, 26(5):1123–1138, 2006.

[9] DoD. *Department of Defense Information Technolgoy Security Certification and Accreditation Process (DITSCAP) Application Manual*. US-Department of Defense, July 2000.

[10] R. Falcone and C. Castelfranchi. Social Trust: A Cognitive Approach. In *Trust and Deception in Virtual Societibes*, pages 55–90. Kluwer Academic Publishers, 2001.

[11] M. S. Feather. Towards a Unified Approach to the Representation of, and Reasoning with, Probabilistic Risk Information about Software and its System Interface. In *Proc. of ISSRE'04*, pages 391–402. IEEE Press, 2004.

[12] M. S. Feather, S. L. Cornford, K. A. Hicks, and K. R. Johnson. Applications of tool support for risk-informed requirements reasoning. *Computer Systems Science & Engineering*, 20(1):5–17, 2005.

[13] R. Fredriksen, M. Kristiansenand, B. A. Granand, K. Stølen, T. A. Opperud, and T. Dimitrakos. The CORAS framework for a model-based risk management process. In *Proc. of SAFECOMP'02*, *LNCS 2434*, pages 94–105. Springer-Verlag, 2002.

[14] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modeling Security Requirements Through Ownership, Permission and Delegation. In *Proc. of RE'05*, pages 167–176. IEEE Press, 2005.

[15] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modelling Social and Individual Trust in Requirements Engineering Methodologies. In *Proc. of iTrust'05*, *LNCS 3477*, pages 161–176. Springer-Verlag, 2005.

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

[17] J. Jacobson. Safety Validation of Dependable Transportation Systems. In *Proc. of ITSC'05*, pages 1–6, 2005.

[18] S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino. A unified framework for enforcing multiple access control policies. In *Proc. of PODS'97*, pages 474–485. ACM Press, 1997.

[19] A. Jøsang and S. Presti. Analysing the Relationship Between Risk and Trust. In *Proc. of iTrust'04*, *LNCS 2995*, pages 135–145. Springer-Verlag, 2004.

[20] S. Lee, R. Gandhi, and G. Ahn. Security Requirements Driven Risk Assessment for Critical Infrastructure Information Systems. In *Proc. of SREIS'05*, 2005.

[21] D. Manchala. Trust Metrics, Models and Protocols for Electronic Commerce Transactions. In *Proc. of ICDCS'98*, pages 312–321. IEEE Press, 1998.

[22] N. Mayer, A. Rifaut, and E. Dubois. Towards a Risk-Based Security Requirements Engineering Framework. In *Proc. of REFSQ'05*, 2005.

[23] D. M. Nicol, W. H. Sanders, and K. S. Trivedi. Model-Based Evaluation: From Dependability to Security. *TDSC*, 1(1):48–65, 2004.

[24] D. Shapiro and R. Shachter. User-Agent Value Alignment. In *Proc. of The 18th Nat. Conf. on Artif. Intell.* AAAI, 2002.

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

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

[27] E. Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, 1996.