UNIVERSITY
OF TRENTO

**DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE**

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

# Security Requirements Engineering via Commitments

Fabiano Dalpiaz, Elda Paja, and Paolo Giorgini

July 2011

Technical Report # DISI-11-469

# Security Requirements Engineering via Commitments

Fabiano Dalpiaz, Elda Paja, Paolo Giorgini
*Dept. of Information Engineering and Computer Science*
*University of Trento, Italy*
{*fabiano.dalpiaz, elda.paja, paolo.giorgini*}*@disi.unitn.it*

***Abstract*—Security Requirements Engineering (SRE) is concerned with the identification of security needs and the specification of security requirements of the system-to-be. Mainstream approaches to SRE either focus on technical security mechanisms or suggest high-level organizational abstractions that are hard to map to the actual design. Social commitments are a simple yet powerful abstraction to model social interactions and can be used effectively to specify security requirements. In this paper, we build on our previous work proposing a novel goal-oriented modelling language called SecCo—Security via Commitments—where the concept of social commitment between social and technical actors is adopted to specify security requirements. Commitments enable the development of robust applications, wherein security needs are satisfied by assigning contractual validity to interactions.**

***Keywords*-Security requirements; Goal models; Commitments**

## I. Introduction

Software systems are subject to security threats which influence organizational assets [1]. *Security requirements* are, therefore, specified and then translated to a set of *security mechanisms* to be developed in the actual system. While some security threats are technological (e.g., distributed denial of service attacks and viruses), others are *social*, as they arise from the interaction between humans/organizations and software, and how information is manipulated.

The importance of considering security from a social and organizational perspective is widely recognized in literature [2]–[5]. However, such approaches still do not characterize high-level organizational security needs in terms of more specific security mechanisms to implement. Solutions either rely on purely technical mechanisms (e.g. [1]), or suggest high-level concepts (e.g. [2], [3]) that are hard to map to technical security requirements.

Analysing security from an organizational perspective means analysing social interactions between actors, their responsibilities, information flow constraints, norms and laws actors should comply with. *Social commitments* are a simple yet powerful abstraction to model social interactions [6]. A commitment is a quaternary relation C(debtor, creditor, antecedent, consequent) in which a debtor agent promises (*commits*) to a creditor agent that, if the antecedent is brought about, the consequent will be brought about. Commitments are purely social abstractions that are rooted in interaction: they are created and they evolve as agents exchange messages. Since they have contractual validity, commitments can be used to build robust applications: non-compliance might lead to further commitments on the part of the violator.

Commitments are an effective means to specify security requirements too. An agent can commit to another for the integrity of a resource, for the non-disclosure of confidential data, for the usage of some resource according to the need-to-know principle, for the redundant fulfilment of a delegated goal, for the non-repudiation of a delegated goal, and so on. These security requirements can be effectively mapped to service interfaces, in which the provider commits to the consumer for the satisfaction of certain security properties while delivering the service.

In this paper, we start from our previous work on (Secure) Tropos [3] and we propose a novel goal-oriented modelling language to specify security requirements via commitments. The language is called *SecCo* (Security via Commitments) and proposes, along with a revised set of high-level organizational concepts from Tropos (i.e., actor, goal, delegation, authorization, ...), the concept of *social commitment* between social and technical actors to specify security requirements. Commitment specifications can be used for the design and the development of applications whose interactions satisfy the security needs.

The paper is structured as follows. Section II presents related work. Section III outlines the SecCo language. Section IV describes the three operational views of SecCo (social, resource, authorization) that enable modelling security needs. Section V introduces the commitments view that specifies security requirements via commitments. Section VI discusses the approach and presents our conclusions.

## II. Related work

The requirements engineering community has acknowledged the importance of considering security since the early stages of software development [7], [8].

In [9], the authors introduce a framework for security requirements based on the notions of delegation and trust of execution / permission. Monitoring is used as an organizational pattern to overcome trust issues. SecCo, instead, ensures security via commitments, concentrating on the interaction between actors.

Secure Tropos [10] models security concerns throughout the whole development process. Security requirements are expressed as *security constraints*, which should be satisfied together with the functional requirements. Potential threats and attacks are considered as well, to analyse and find the best way to overcome possible vulnerabilities. SecCo separates security needs from security requirements, and binds security to interaction via commitments.

Abuse cases [11] extend use cases to capture and analyse security requirements. An abuse case specifies a type of interaction between a system and one or more actors, where the results of the interactions are negative/harmful. It includes a range of security concerns that might be abused, as well as a description of the harm that might be caused. In a similar spirit, misuse cases [12] exploit use cases to represent sequences of actions that a system or other entities can perform, interacting with *misusers* of the entity and causing harm if the sequence is allowed to complete. These approaches exploit negative scenarios to elicit and analyse security requirements. SecCo focuses on how actors should interact, and defines a set of commitments that protects their interaction. The approaches are complementary.

Lamsweerde [5] deals with security engineering at the application layer. Security requirements are specified by two models: a model of the system-to-be and an anti-model. The anti-model includes vulnerabilities and capabilities needed to achieve the anti-goals of the security goals (from the former model) that are endangered. Anti-goals are refined in threat trees, whose leaf nodes represent either vulnerabilities observable by the attacker or anti-requirements implementable by the attacker. Differently, SecCo captures security at the organizational level.

Liu et al. [2] present a goal-oriented methodology based on *i** to deal with security and privacy requirements. Security dimensions are modelled as softgoals, and security requirements analysis is performed to verify whether the system is secure. Analysis identifies potential system attackers/abusers, vulnerabilities (propagated along dependency links), thereby suggesting countermeasures. Their solution falls short when considering security issues through the later phases of the development process [10].

Elahi's work [13] extends the *i** framework by supporting security trade-off analysis. The authors propose a conceptual modelling technique to reach a good enough security level in a multi-actor setting. This technique offers the possibility to assess the impact of assessing security mechanisms on actors' goals and threats. Vulnerabilities refer to the deficiencies in the structure of goals and activities of intentional agents. Unlike SecCo, they do not take into account vulnerabilities related to actors interaction.

Haley et al. [4] define security requirements as constraints over functional requirements. They consider context as an important factor having a deep effect on security requirements. Moreover, a structure of satisfaction arguments is employed to verify the correctness of security requirements. SecCo considers security earlier, at the organizational level, and binds security to interaction.

Breaux and Antón [14] present a methodology to systematically extract security (legal) requirements from regulatory texts. They acquire and present data requirements, thereby assigning priorities to them, to ensure law compliance and avoid inappropriate information disclosure. Though relying on contractual rules, they focus only on data usage.

## III. SecCo modelling language: overview

We provide an outline of SecCo (Security Commitments), our modelling language for SRE. Like other goal-oriented approaches to SRE, e.g. [2], [3], [10], SecCo describes the organization in terms of *intentional actors* (i.e. having goals). The actors we consider are also *social*: they depend one on another for the fulfilment of their respective goals. Actor intentionality and sociality are supported by the social view (IV-A). SecCo enables to express *security needs* to constrain how interaction takes place. For instance, an actor might want to guarantee the confidentiality of an exchanged resource, or redundant fulfilment of a delegated task.

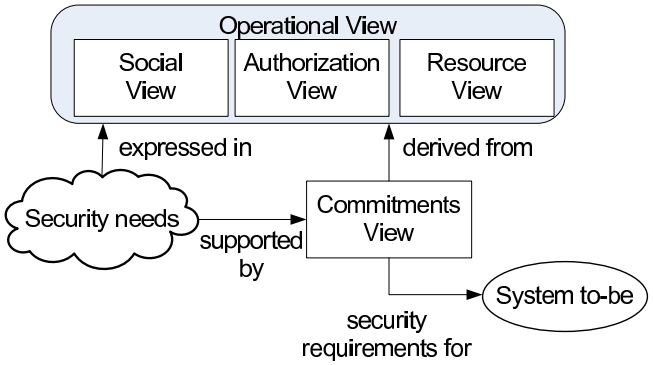


Figure 1. From the operational view to security requirements

Actors use and exchange resources to fulfil their goals. SecCo's resource (Section IV-B) and authorization (Section IV-C) views support an elaborated characterization of resources and distinguish between the actual usage of resources and the granted authorizations.

Identifying security needs and discovering potential breaches is not sufficient to successfully complete security requirements analysis. The missing step is specifying security requirements that, if implemented, satisfy the security needs. To this extent, SecCo relies on the concept of commitment. Security requirements are a set of commitments between actors (Section V). Such commitments shall be established—via security mechanisms—and continuously monitored. Actors can make commitments to ensure redundancy, integrity, non-disclosure, need-to-know, etc.

Figure 1 outlines SecCo. *Security needs* are a key concept: they are *expressed in* the operational view that describes

the setting and are *supported by* the commitments view. The operational view consists of three views: social, authorization, and resource. Together, these views provide a comprehensive picture of the setting which includes both business concerns and security aspects. The commitments view specifies the security requirements for the system to-be; it is automatically derived from the operational view.

**Running example.** We consider the compliance of Italian public administrations, such as universities, to Italian security and privacy legislation [9]. This law/act specifies requirements over the public administrations to devise internal regulations and policies, based on the ISO-17799 standard, that regulate personal data usage, update, modification and production. The University of Trento (UniTn) has enforced the Data Protection Act since January 14th, 2002.

UniTn offers several international programmes that attract a large number of international students. Suppose an international *student* needs a document from the *programme coordinator*; such document has to be presented to the local immigration office to get his stay permit extended. The following roles are involved:

- *Student*: needs an official document to prove he is enrolled in the study programme and his incomes are enough to afford the stay. He asks the programme coordinator to issue the document. For this reason, he has to provide his personal data, as well as financial information. His personal data is stored in the UniTn information system.
- *Programme Coordinator*: issues the official document for the student. He might transfer responsibility for parts of this activity to his secretary.
- *Secretary*: retrieves student information (personal data and financial data) from the information system and drafts the document.
- *IS Manager*: manages the information about students stored in the UniTn information system in accordance with confidentiality restrictions.

## IV. Modelling security needs

We detail the three sub-views that constitute the operational view of SecCo. Together, these views enable modelling the security needs expressed by stakeholders.

### A. Social view

The social view builds on top of existing goal-oriented languages for SRE, in particular SI* [9]. Our aim is to stay with a minimal and consistent set of concepts that can be effectively used to depict the operational aspects of the considered setting. Figure 2 illustrates the social view on the running example.

We consider an abstract concept of actor, and refine it to two distinct concrete concepts: role and agent. Agents play (adopt) roles at runtime, and they can change the roles they play. Some agents are known since requirements-time. For instance, the prefecture of Trento is an agent, for students should invariably interact with it to renew their stay permit.

An actor *wants* to achieve one or more goals, and has capabilities to fulfil some of them without interacting with others. A goal can be AND/OR decomposed to two or more subgoals. In an AND-decomposition (OR-decomposition), the parent goal is achieved if all (at least one) subgoals are satisfied. Goals can contribute to one another. We support two types of full contribution. In positive (negative) contribution, the satisfaction of one goal gives evidence for the satisfaction (denial) of the contributed goal [15]. In Figure 2, the secretary wants to achieve goals “write new document”, “get student records”, etc. She has capability for “get student records”, which is AND-decomposed to two sub-goals.

We tie together goals and resources in various ways:

- an actor *possesses* (disposes of) a set of resources;
- an actor *needs* one or more resources to fulfil a goal;
- an actor *produces* resources while fulfilling a goal;
- an actor *modifies* a resource while fulfilling a goal. A resource is modified if, despite of the change or update, the resource identity is unvaried. For example, the personal data file of a student can be modified if the student’s address changed.

In Figure 2, the secretary’s goal “Write new document” produces an “Official document” for the student and needs resource “Document template”. The secretary *possesses* resource “Document template”.

We consider social actors that collaborate to fulfil their own objectives. SecCo supports two types of social relationship: *goal delegation* and *resource provision*. Whereas the former captures the expectations of one actor on others (the goals he delegates), the latter represents the exchange of resources among actors.

A key concept in the social view is that of *security need*. This term refers to the expectations concerning security that actors impose on the social relationships they participate in.

**Goal delegation**. A delegator actor delegates the fulfilment of a goal (delegatum) to a different delegatee actor. In Figure 2, the student delegates the fulfilment of goal “Write document for immigration office” to the programme coordinator. Delegations can have a set of security needs that involved actors should preserve. Some of these needs are the following:

- *Non-repudiation* (NonRep): the delegator actor wants the delegatee actor not to be able to challenge the validity of the goal delegation. A non-repudiation security need requires the adoption of security mechanisms that guarantee the delegatee cannot repudiate the delegation. As we will detail in Section V, such security solution consists of the establishment of a commitment—for the non-repudiation of that goal delegation—from the delegatee to the delegator. For instance, the programme coordinator wants non-repudiation for the delegation of his goal “Write new document” to the secretary.

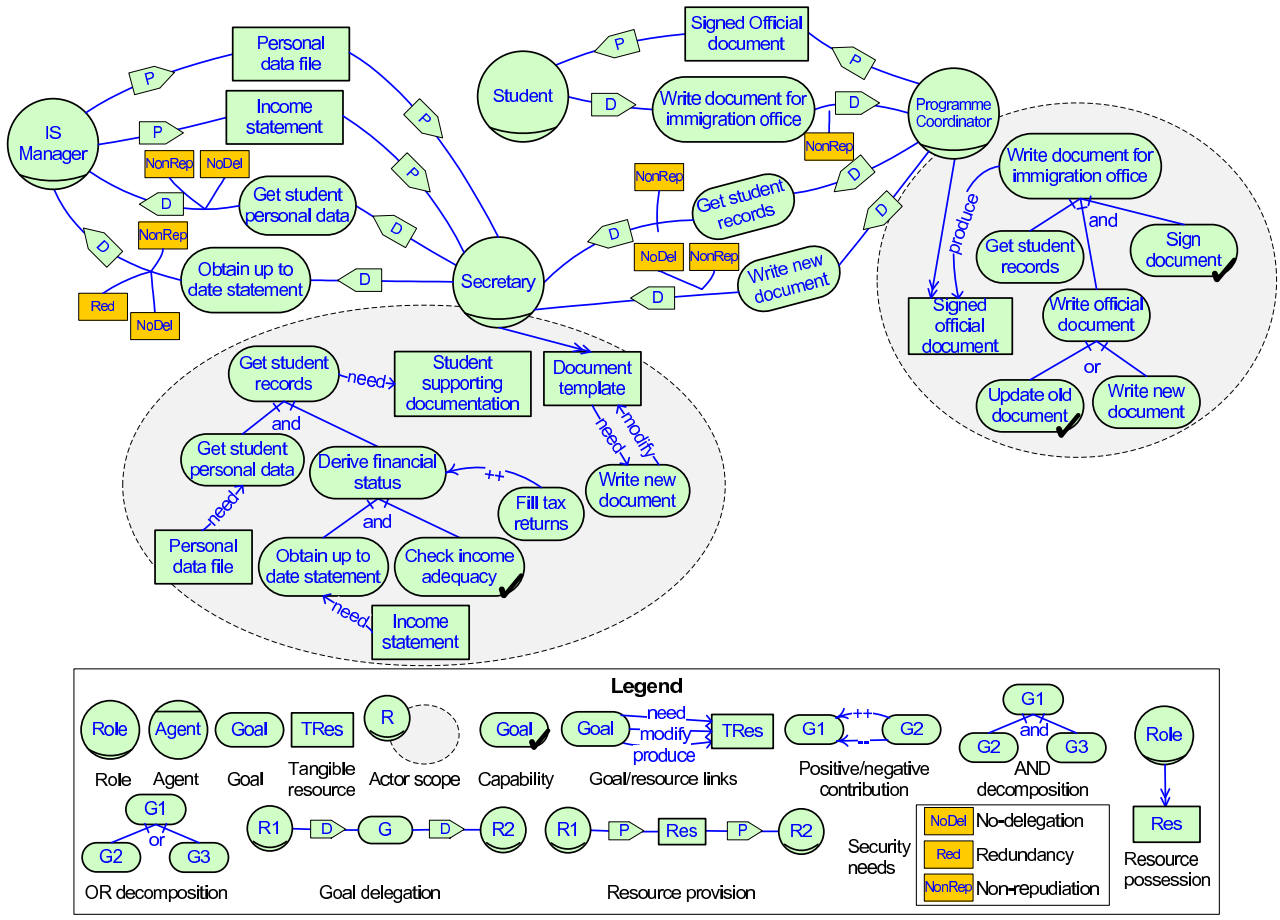


Figure 2. Social view for the stay permit scenario

- *Redundancy* (Red): the delegatee has to adopt redundant strategies for the achievement of the delegated goal. He can either use different internal capabilities, or can rely on multiple actors. To guarantee such security need, the delegatee has to make a commitment to the delegator for redundant fulfilment of the goal.
- *No-delegation* (NoDel): the delegator wants the delegatee not to further delegate goal fulfilment. No-delegation is closely related to *trust*: the delegator trusts *that* specific delegatee for some goal, and does not trust other actors the delegatee might want to involve. Such security need implies a commitment from the delegatee to the delegator: the delegatee promises not to further delegate the fulfilment of that goal. For example, the secretary wants the IS Manager not to delegate goal "Get student personal data"; she might fear someone else would violate data confidentiality.

**Resource provision**. This relationship specifies the exchange of tangible resources (TResource) between actors. Intangible resources (e.g. ideas) cannot be transferred unless made concrete by a tangible means (e.g. a paper, an e-mail). We further elaborate on this distinction in Section IV-B.

Resource provision can be subject to security needs that restrict the usage of received resources. SecCo consider these needs by combining its three operational views. This will become clearer in Section IV-C.

### B. Resource view

Resources play a key role in the social view: actors possess resources as well as they use, modify, produce, and distribute them while fulfilling their goals. The purpose of SecCo's resource view is to devise adequate modelling primitives to characterize resources. We consider only informational resources.

Similarly to [4], a resource can be tangible (TResource) or intangible (IResource). Tangible resources reflect the concrete entities (including electronic ones, such as e-mails) that actors exchange (via resource provision). Intangible resources reflect the informational content that actors intend to transfer by exchanging tangible resources. Intangible resources are exchanged only when *madeTangibleBy* a tangible resource. For instance, in Figure 3, the "Financial status" of the student is an intangible resource (it exists irrespective

of any tangible resource representing it). Such information can be transferred only if made tangible; for example, when represented by a printed “Income statement”.

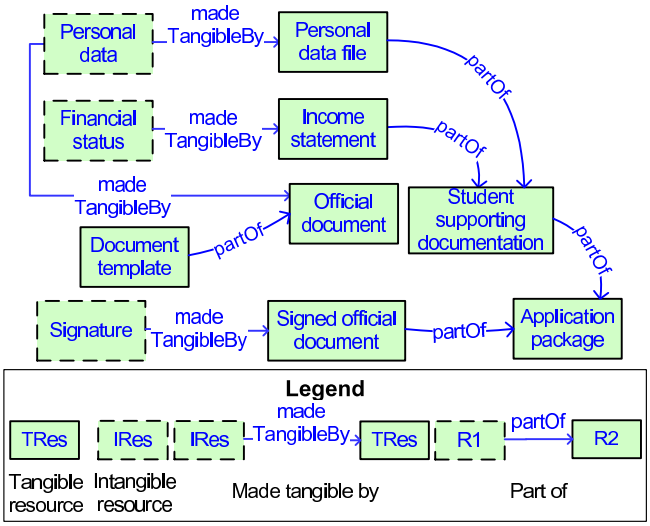


Figure 3. Resource view for the stay permit scenario

Another feature of the resource view is to support composite resources. We enable that by means of the *partOf* relation, which can be applied between homogeneous resources (tangible to tangible, intangible to intangible). This allows for representing that a “signed official document” is part of the “application package” the student should deliver.

The resource view is flexible in representing resources and the relations between them:

- An intangible resource can be made tangible by different tangible resources. For instance, “Personal data” is made tangible by both “Personal data file” and “Official document”.
- A tangible resource can have no relevant intangible resource. For instance, “Document template” contains no relevant information concerning the issuing of a permit of stay for an international student.
- A tangible resource might be part of multiple tangible resources. Though not in Figure 3, an “Income statement” might be part of a scholarship application too.

### C. Authorization view

An adequate representation of authorizations is necessary to determine if resources are exchanged and used in compliance with confidentiality restrictions. The resource owner is the unique actor that can legitimately transfer rights to other actors. However, he might transfer full rights to another actor, so that the latter becomes entitled to transfer the same rights the owner can grant.

An actor owns an arbitrary number of intangible resources. We do not take into account resources with multiple owners here. We support the transfer of rights between two actors via *delegation of authority*. An actor can grant/receive an arbitrary number of delegations of authority. Authority can be specified along three dimensions:

- *Scope*: authority over resources can be limited to their usage in the scope of a specific purpose (i.e. certain goals). In SecCo, if a goal is in the scope, all its sub-goals—according to the delegator’s goal model—are in scope too.
- *Operations*: transferred rights relate to different operations/actions an actor can perform on the resources. In SecCo, we support four basic operations: usage, modification, production, and distribution. We do not consider revocation of permissions in this paper. The four supported operations are directly linked to the way resources are manipulated by actors in the social view. Authority of usage goes in parallel with the *needs* relation, authority of modification with *modifies*, authority of production with *produces* relation, and authority of distribution with *resource provision*.
- *Authority to delegate*: when the actor receiving the authority can further delegate such authority to other actors. In SecCo, we support a special kind of authority called AuthorityToDelegate (see [16]). This is a stronger authority that includes not only the permission to perform operations, but also that of further propagating rights over those resources to other actors. Such further delegation should, however, be compatible with the authority scope the delegator is granted.

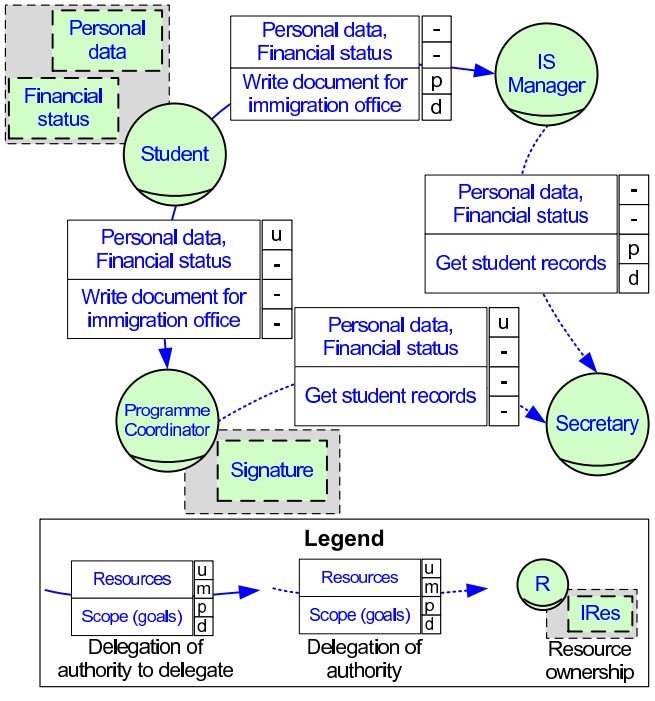


Figure 4. Authorization view for the stay permit scenario

Figure 4 shows the authorization view for the stay permit

scenario. The student owns his "Personal data" and "Financial status". The white boxes on top of arrows are authorizations. Depending on the arrow line, authorization to delegate is granted (full line) or not (dotted line). An authorization box contains three slots: the upper slot is the list of resources over which authorization is delegated; the lower slot is the scope; and the right slot defines the allowed operations (from top to bottom: use, modify, produce, distribute). The student authorizes the usage of personal data and financial status to the programme coordinator in the scope of goal "Write document for immigration office". Since authority to delegate is transferred, the programme coordinator delegates authority to use personal data and financial status to the secretary in the scope of goal "Get student records" (which is a sub-goal of "Write document for immigration office"). Authority to delegate is not transferred to the secretary.

The authorization view expresses security needs on the use of resources. Some of these needs are the following:

- *Non-disclosure*: when authority is granted without transferring authority to delegate. An actor grants another the authority to perform some operations on a resource (any combination of use, modify, produce, distribute), as long as the resource is not disclosed to unauthorized actors. For example, the IS Manager expresses such security need in the authorization over resources personal data and financial status granted to the secretary.
- *Need-to-know*: when the transfer of authority to delegate is restricted to a goal scope. The actor granting the authority enables the delegatee to delegate permission to others as long as other actors conduct operations on the resource within the specified scope. The student's authorization to the IS Manager expresses a need-to-know security need: personal data and financial status should be produced or distributed in the scope of goal "Write document for immigration office".
- *Integrity*: when the authority to modify is not granted to the delegatee. The IS Manager expresses such security need on the delegation of authority over resources personal data and financial status to the secretary.

## V. Specifying security requirements via commitments

The operational view described in the previous sections models business aspects of the considered setting as well as security needs. As shown in Section IV-C, however, security needs are often modelled implicitly. Thus, security requirements engineering might be unaware of these needs and of the security requirements they imply.

SecCo goes one step further with its commitments view. This view is *automatically derived* from the operational view and contains a high-level specification of the *security requirements*—expressed via social commitments—that, if actors comply with, satisfies the security needs. Though not detailing automated transformation rules, we provide here the intuition behind the mapping between the security needs in the operational view and the commitments view.

An important feature of SecCo is to relate security requirements to *interaction* between actors. However, unlike technical approaches to computer security, interaction is understood in business terms. At requirements time, commitments are expressed at the level of roles (with the exception of the agents that are already known). At runtime, these commitments shall be made by the involved agents (playing those roles). After their identification, it is therefore crucial, during the architectural design phase, to link commitments to technical security mechanisms that guarantee their satisfaction.

We specialize the notion of commitment proposed by Singh [6], so that it can be exploited in the context of security requirements. In SecCo, a commitment is made by a debtor actor to a creditor actor for the satisfaction of a security need. In turn, security needs are defined in terms of the concepts used in the operational view (as shown in the previous sections).

The way commitments are implemented is highly dependent on whether the involved actors are agents or roles. If the debtor is a role, making that commitment becomes a necessary condition for any agent playing that role, that has to make such commitment to adopt the role. In other words, the commitment becomes part of the description of the role. If the creditor is a role, the commitment is a security guarantee for any agent playing that role while interacting with the debtor. If the debtor is an agent, the system to-be should ensure that the specific agent makes those security commitments when interacting with others. If the creditor is an agent, such commitments become prerequisites for other agents interacting with it.

| Id | Commitment type |
|---|---|
| (a) | $\mathsf{C}(a, b,$need-to-know$(\mathcal{R}, \mathcal{G}, Ops))$<br>Actor $a$ commits to actor $b$ that resources in $\mathcal{R}$ will be used/modified/produced/distributed (as specified in *Ops*) *only* in the scope of the goals in $\mathcal{G}$ |
| (b) | $\mathsf{C}(a, b,$non-disclosure$(\mathcal{R}))$<br>$a$ commits to $b$ that resources in the set $\mathcal{R}$ will not be distributed to unauthorized actors |
| (c) | $\mathsf{C}(a, b,$integrity$(\mathcal{R}))$<br>$a$ commits to $b$ that resources in $\mathcal{R}$ will not be modified (integrity will be preserved) |
| (d) | $\mathsf{C}(a, b,$non-repudiation$(\mathcal{G}))$<br>$a$ commits to $b$ that he will not repudiate that $a$ has been delegated the goals in $\mathcal{G}$ |
| (e) | $\mathsf{C}(a, b,$redundancy$(\mathcal{G}))$<br>$a$ commits to $b$ that redundant strategies will be adopted to fulfil the goals in $\mathcal{G}$ |
| (f) | $\mathsf{C}(a, b,$no-delegation$(\mathcal{G}))$<br>$a$ commits to $b$ that goal $\mathcal{G}$ will not be delegated to others |

Table I
COMMITMENT TYPES TO EXPRESS SECURITY REQUIREMENTS

Table I shows how the security needs expressed in the

| Id | Debtor | Creditor | Security Requirement |
|---|---|---|---|
| $C_1$ | IS Manager | Student | need-to-know(personal data $\wedge$ financial status, write document for immigration office, p $\wedge$ d) |
| $C_2$ | Progr. Coord. | Student | need-to-know(personal data $\wedge$ financial status, write document for immigration office, u) |
| $C_3$ | Secretary | Progr. Coord. | need-to-know(personal data $\wedge$ financial status, get student records $\wedge$ write new document, u) |
| $C_4$ | Secretary | IS Manager | need-to-know(personal data $\wedge$ financial status, get student records, p $\wedge$ d) |
| $C_5$ | Secretary | IS Manager | non-disclosure(personal data $\wedge$ financial status) |
| $C_6$ | IS Manager | Student | integrity(personal data $\wedge$ financial status) |
| $C_7$ | Progr. Coord. | Student | integrity(personal data $\wedge$ financial status) |
| $C_8$ | Secretary | Progr. Coord. | integrity(official document) |
| $C_9$ | Secretary | IS Manager | integrity(personal data $\wedge$ financial status) |
| $C_{10}$ | Progr. Coord. | Student | non-repudiation(write document for immigration office) |
| $C_{11}$ | Secretary | Progr. Coord. | non-repudiation(write new document $\wedge$ get student records) |
| $C_{12}$ | IS Manager | Secretary | non-repudiation(get student personal data $\wedge$ obtain up to date statement) |
| $C_{13}$ | IS Manager | Secretary | redundancy(obtain up to date statement) |
| $C_{14}$ | Secretary | Progr. Coord. | no-delegation(write new document) |
| $C_{15}$ | IS Manager | Secretary | no-delegation(get student personal data $\wedge$ obtain up to date statement) |

Table II
SECURITY REQUIREMENTS EXPRESSED VIA COMMITMENTS IN THE STAY PERMIT SCENARIO

operational view lead to specific commitments in the commitments view. Table II lists the commitments for the stay permit scenario derived from the operational view presented in the previous sections. the The semantics of the various commitment types in Table I is as follows:

(a) A need-to-know commitment from $a$ to $b$ implies that a set of resources $\mathcal{R}$ will be used / modified / produced / distributed (in accordance with the operations specified in *Ops*) only within the scope of a set of goals $\mathcal{G}$. In case the committed actor has the authority to delegate rights, other actors might be in turn authorized for the resource. However, to guarantee the commitment made by $a$, each of them has to make a commitment to $b$ for the need-to-know of the resources. For example, in Table II, the IS Manager commits ($C_1$) to the student for the need-to-know of personal data and financial status in the scope of goal "Write document for the immigration office". Allowed operations are production and distribution. In turn, this implies a commitment ($C_4$) from the secretary to the IS Manager for the same resources and operations in the scope of the sub-goal "Get student records".

(b) A non-disclosure commitment says that the debtor will not distribute some resources to unauthorized actors. This type of commitment protects delegations of authority that include resource distribution but not the authority to delegate such permission. For example, the secretary commits ($C_5$) to the IS Manager for the non-disclosure of personal data and financial data.

(c) An integrity commitment for some resources $\mathcal{R}$ implies that these resources will not be modified. The debtor actor commits that not only he will not modify the resource, but also that—if he distributes such resource to other actors—each of these actors will commit for the integrity of the resource. For example, the programme coordinator commits ($C_7$) to the student for the integrity of personal data and financial status, since he gets no authority to modify such data. In turn, a similar commitment ($C_8$) is made from the secretary to the programme coordinator.

(d) Commitments for non-repudiation are essential to support accountability. We are concerned here with non-repudiation of goal delegations. The committed actor promises he will not repudiate that he was delegated the fulfilment of the goals in $\mathcal{G}$. For example, the programme coordinator commits ($C_{10}$) to the student for the non-repudiation of goal "Write document for immigration office".

(e) A commitment for redundant goal fulfilment says that the debtor will fulfil the goals in $\mathcal{G}$ by adopting redundant strategies. External actors can be involved too. However, the same goal cannot be delegated twice to the same actor, as that would not ensure redundancy. Redundancy commitments support reliability. For example, the IS Manager commits ($C_{13}$) for redundant fulfilment of goal "Obtain up to date statement". The IS Manager can fulfil it by either retrieving two statements from different databases, or delegating the task to two technicians, or retrieving a statement from a database and delegating to a technician.

(f) A no-delegation commitment tells that a debtor will fulfil a goal without further delegations. Such restriction applies to the descendants of the goal in the goal tree. The IS Manager commits ($C_{15}$) to the secretary that he will not delegate goals "Get student personal data" and "Obtain up to date statement" to others.

**Operationalizing commitments.** Security commitments are security requirements at the organizational level. At the technical level, they result in operationalization via security mechanisms that ensure commitments to be satisfied.

Commitment $C_1$ requires to ensure need-to-know. A possible security mechanism for $C_1$ is to log access to the information system and require IS users to specify which is the purpose for which they access confidential data. The purpose might be inferred from interaction. In our example, the system to-be can check if the IS manager is using personal data and financial status upon a request (e.g. by the secretary) for writing the document for the immigration office. $C_6$ is about integrity. At least two technical options exist: preventively denying modification grants to the IS Manager, or monitoring its access to personal data and financial status.

$C_{10}$ is about non-repudiation of goal "Write document for immigration office". An information system can be developed: students delegate this goal through the IS, and the IS Manager has to accept the task. The log of the information system is the proof that the delegation was accepted. To implement commitments $C_5$ (non-disclosure) and $C_{15}$ (non-repudiation), the information flow should be tracked. While $C_5$ directly refers to resources, $C_{15}$ does it indirectly, since delegated goals produce resources that can be tracked.

## VI. Discussion and Conclusion

In this paper we have presented SecCo, a novel goal-oriented modelling language for security requirements engineering. SecCo covers both the analysis of *security needs*—in its operational view—and the derivation of *security requirements*—in its commitments view—that should be implemented to satisfy the needs.

SecCo specifies requirements via social commitments between actors, thereby relating security to interactions among actors. The commitments view is automatically inferred from the operational view, which consists of three views that enable requirements engineers to model orthogonal aspects of the considered setting. We exploit a non-redundant set of concepts that allows for focussing on the most important security concerns at the requirements level.

This paper puts the basis for several research threads. We are particularly interested in using SecCo to design composite services. To such extent, we plan to: (i) formalize the automated derivation of the commitments view from the operational view; (ii) define commitments operationalizations that detail how security requirements are fulfilled (for instance, via SLAs); (iii) devise a supporting methodology for SecCo; (iv) validate the approach on industrial case studies (from the EU-sponsored project Aniketos).

## References


[1] D. G. Firesmith, "Security Use Cases," *Journal of Object Technology*, vol. 2, no. 3, pp. 53–64, 2003.

[2] L. Liu, E. Yu, and J. Mylopoulos, "Security and Privacy Requirements Analysis within a Social Setting," in *Proceedings of the 11th IEEE International Conference on Requirements Engineering (RE 2003)*. IEEE Computer Society, 2003, pp. 151–161.

[3] P. Giorgini, F. Massacci, and J. Mylopoulos, "Requirement Engineering meets Security: A Case Study on Modelling Secure Electronic Transactions by VISA and Mastercard," in *Proceedings of the 22nd International Conference on Conceptual Modeling (ER 2003)*, ser. LNCS, vol. 2813. Springer, 2003, pp. 263–276.

[4] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh, "Security Requirements Engineering: A Framework for Representation and Analysis," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 133–153, 2008.

[5] A. van Lamsweerde, "Elaborating Security Requirements by Construction of Intentional Anti-Models," in *Proceedings of the 26th International Conference on Software Engineering (ICSE 2004)*. IEEE Computer Society, 2004, pp. 148–157.

[6] M. P. Singh, "An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts," *Artificial Intelligence and Law*, vol. 7, no. 1, pp. 97–113, 1999.

[7] P. Devanbu and S. Stubblebine, "Software Engineering for Security: a Roadmap," in *Proceedings of the Conference on The Future of Software Engineering (FOSE 2000)*, 2000, pp. 227–239.

[8] E. Dubois and H. Mouratidis, "Guest Editorial: Security Requirements Engineering: Past, Present and Future," *Requirements Engineering*, vol. 15, no. 1, pp. 1–5, 2010.

[9] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Modeling Security Requirements through Ownership, Permission and Delegation," in *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE 2005)*. IEEE Computer Society, 2005, pp. 167–176.

[10] H. Mouratidis and P. Giorgini, "Secure Tropos: A Security-Oriented Extension of the Tropos methodology," *International Journal of Software Engineering and Knowledge Engineering*, vol. 17, no. 2, pp. 285–309, 2007.

[11] J. McDermott and C. Fox, "Using Abuse Case Models for Security Requirements Analysis," in *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99)*. IEEE Computer Society, 1999, pp. 55–64.

[12] G. Sindre and A. L. Opdahl, "Eliciting Security Requirements with Misuse Cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, 2005.

[13] G. Elahi and E. Yu, "A Goal Oriented Approach for Modeling and Analyzing Security Trade-offs," in *Proceedings of the 26th International Conference on Conceptual modeling (ER 2007)*, ser. LNCS, vol. 4801, 2007, pp. 375–390.

[14] T. D. Breaux and A. I. Antón, "Analyzing Regulatory Rules for Privacy and Security Requirements," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 5 –20, 2008.

[15] P. Giorgini, J. Mylopoulos, and R. Nicchiarelli, Eleonora and-Sebastiani, "Reasoning with Goal Models," in *Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002)*, 2002, pp. 167–181.

[16] J. D. Moffett and M. S. Sloman, "Delegation of Authority," *Integrated Network Management II*, pp. 595–606, 1991.