Security Requirements Engineering for Service-Oriented Applications

Fabiano Dalpiaz, Elda Paja, Paolo Giorgini

University of Trento - DISI, 38123, Povo, Trento, Italy
{fabiano.dalpiaz, paja, paolo.giorgini}@disi.unitn.it

Abstract. Security Requirements Engineering (SRE) is concerned with detecting and analysing security issues early in the software development process. Some variants of *i** start since early requirements and rely on modelling actors and their dependencies. Though useful for traditional information systems development, these approaches adopt a bird's eye perspective that is inadequate for service-oriented applications, in which multiple autonomous and heterogeneous agents interact to achieve their own strategic interests.

In this paper we present SecCo (Security via Commitments), a novel SRE framework expressly thought for service-oriented settings. The key intuition is to relate security requirements to interaction. In order to do so, we specify security requirements in terms of social commitments, promises with contractual validity between agents. These commitments describe the security properties the service provider commits to ensure to the consumer while delivering the service.

1 Introduction

Software systems are subject to security threats, which may influence organisational assets. Thus, the specification of *security requirements* as well as their implementation by means of *security mechanisms* are of utmost importance. Many security threats are not technical; rather, they are *social*, as they originate from the interactions between social actors (humans and organisations).

Several proposals consider social threats by modelling and analysing security since the early requirements phases. Many frameworks extend *i** [1–3] to model the environment in terms of social actors and their dependencies. Though they represent dependencies between actors, these approaches adopt a centralised view on the modelled setting, which leaves little space to the autonomy and heterogeneity of the actors. Indeed, they implicitly assume that participating actors will behave as described in the goal models.

Service-oriented computing enables cross-organisational business processes and, more generally, black-box interactions among autonomous actors (on the basis on service interfaces). In our previous work [4], we revisited service-orientation in terms of goal-oriented agents that adopt specific roles and interact with others by making and getting social commitments [5]. A commitment is a quaternary relation C(debtor, creditor, antecedent, consequent) in which the debtor promises (*commits*) to the creditor that, if the antecedent is brought about, the consequent will be brought about. As soon as the antecedent holds, the debtor becomes unconditionally committed to bring about the consequent. Unlike dependencies, commitments are a purely social abstraction. Dependencies imply the intention on the part of the dependee to bring about the dependum [6]. Commitments, on the other hand, imply no intention, they exist as a consequence of the interaction between the debtor and the creditor agent [5]. A service interface is a set of commitments the service provider makes to prospective service consumers.

This paper introduces SecCo [7] (Security via Commitments), a novel SRE framework that combines the early-requirements perspective of SI* [2] with ideas from service-oriented applications [4]. The key idea is to relate security requirements to *interaction*. This means adding constraints to the way actors exchange resources, and to the delegation of responsibilities. These constraints are commitments the actors shall comply with while interacting. For example, a service provider could commit to the privacy of the consumer's personal data, which is required as input to the provided service. Again, the same service provider may commit to the non-delegation of a service to other actors.

2 Objectives of the research

The main objective of this research thread is to support security requirements engineering in the context of service-oriented settings. Service orientation is becoming a popular paradigm, especially for cross-organisational settings. The analysis of security aspects is of utmost importance, since information is disclosed (and tasks are executed) beyond the "safe" boundaries of a single organisation. Our goal is to devise a modelling framework as well as algorithms that enable the automated discovery of security issues in the models. We are mainly concerned with composite services, which comprise multiple services that relate different actors acting both as service consumers and providers.

Additionally, we aim to derive security requirements that can be effectively used to specify the service under development. Our purpose is to express security requirements within service interfaces. This ensures that the security needs expressed by the stakeholders result in actual commitments the provider makes to the consumer to satisfy these constraints (the security needs) while delivering the service. For instance, if consumers are concerned with the disclosure of personal data, the service interface may declare that the data will not be disclosed to other actors. Irrespective of the service implementation, such interface makes the provider committed for non-disclosure.

3 Scientific contributions

Figure 1 outlines SecCo and shows how security requirements for the composite service under design are derived from the security needs expressed by the stakeholders. SecCo allows to model interaction among actors from different perspectives (views). Security needs are *expressed in* the business view that describes multiple orthogonal perspectives over the considered setting. SecCo currently includes three views: social, authorisation, and resource. We provide more details about these views in Section 3.1. Together, these views provide a comprehensive picture of the setting which includes both business concerns and security aspects.

Security needs are *supported by* the commitments view, which consists of a set of commitments between actors. Such view is a high-level specification of the security

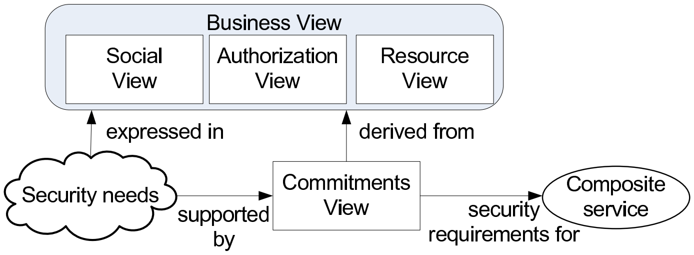

Fig. 1. Outline of SecCo: from security needs to security requirements

requirements for the system-to-be. As long as the actors do not violate those commitments, the security needs expressed by the interacting parties are ensured. The link between goals and commitments has already been discussed in [4], in this work we will concentrate on how commitments can guarantee the security needs. The commitments view can be automatically derived from the business view.

3.1 Multi-view modelling

SecCo relies on multiple views of the same model, each representing a specific perspective on the analysed setting. Multi-view modelling promotes modularity and separation of concerns. Currently, SecCo includes three views:

- *Social view*: a variant of traditional *i**-based frameworks, more specifically of SI*. As such, it models actors and their dependencies. SecCo supports two types of actors: *agent* and *role*. An agent can play multiple roles. Each actor is characterised in terms of *goals* he wants to achieve. Goals can be *AND/OR-decomposed*. Differently from SI*, in which resources are a means to achieve a goal, here goals are linked to *tangible resources*—information represented on some support means—in various ways: a goal can *read*, *produce*, *modify*, or *distribute* a resource. Resource *possession* indicates that an actor can dispose of certain resources without interacting with other actors. There are two social relationships: *resource provision*—representing the exchange of tangible resources—and *goal delegation*. The distinguishing feature of our language is that in the social view one can express security needs by means of annotations associated to elements of the model (e.g. delegation). We discuss the supported security needs in Section 3.2.
- *Resource view*: focuses on the way resources are structured. We distinguish between tangible resources (introduced in the social view) and *intangible resources*, which denote information irrespective of its representation. For instance, Jim's birthday is an intangible resource. Resources can be hierarchically structured via the *part-of* relation, which relates homogeneous resources (intangible to intangible, tangible to tangible). Intangible resources are *made tangible by* tangible resources. For example, Jim's birthday is made tangible by his identity card.
- *Authorisation view*: represents the authorisations actors grant one to another. We distinguish between delegation of *authority* and delegation of the *authority to delegate*. The second type implies that the delegatee can further delegate the received

authorisation. Authorisations are granted by an actor to another for one or more intangible resources, and specific *operations* are authorised: read, produce, modify, and distribute. An authorisation can be limited to a *scope*—a set of goals—that determines the purposes why the delegatee can use tangible resources that represent those intangible resources.

3.2 Expressing security needs

We outline the security needs supported by the SecCo business view with the aid of a small scenario concerning stay permits for international students.

Scenario. An international *student* enrolled at the University of Trento needs a stay permit. To obtain it, he needs an official document to prove his enrolment and that 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 university information system. The *programme coordinator* delegates his task to the *secretary*. She retrieves personal data from the information system and drafts the document, then gives the draft to the programme coordinator who has to sign it. The *IS manager* manages authorisations in the university information system in accordance with confidentiality restrictions.

SecCo currently supports the following security needs:

- *Non-repudiation*: in a goal delegation, the delegator wants to prevent the delegatee from challenging the validity of the delegation (repudiating the delegation). For instance, **(Ex1)** the programme coordinator wants to ensure non-repudiation for the delegation of goal "Write document" to the secretary.
- *Redundancy*: in a delegation, the delegator wants the delegatee to adopt redundant strategies for the achievement of the goal. He can either use different internal capabilities, or can rely on multiple actors. For example, **(Ex2)** the secretary wants the IS manager to adopt redundant strategies to obtain the student's income statement, since provided data is often inconsistent.
- *No-delegation*: the delegator requires the delegatee not to further delegate goal fulfilment. This security need is closely related to *trust*: the delegator trusts *that* specific delegatee for some goal, and does not trust other actors. For example, **(Ex3)** the secretary wants the IS Manager not to delegate goal "Get student personal data", as she is afraid someone else would violate data confidentiality.
- *Non-disclosure*: authority over a resource is granted without transferring authority to delegate. For example, **(Ex4)** the IS Manager authorises the secretary for resources personal data and financial status, but requires non-disclosure of such data.
- *Need to know*: when the granted authority to delegate is limited to a goal scope. The actor granting the authority enables the second actor to delegate permission to others as long as other actors conduct operations on the resources within the specified scope. For example, **(Ex5)** the student authorises the IS Manager on a need-to-know basis: personal data and financial status should be produced or distributed in the scope of goal "Write document for immigration office".
- *Integrity*: an actor does *not* delegate the authority to modify a resource. For example, **(Ex6)** the IS Manager authorises the secretary for personal data and financial status as long as these resources are not modified.

3.3 Deriving security requirements as commitments

SecCo represents security requirements as commitments between actors. In particular, these commitments are between roles, implying that the actual agents playing those roles are expected to make and comply with those commitments. These commitments are created as a consequence of the security needs expressed by the roles while interacting, namely for each security need expressed by a role on an interaction with another, a commitment in the opposite direction will be created. If all agents playing those roles comply with their commitments [4], the security needs will be guaranteed.

Security requirements are automatically derived from the business view. We sketch some security requirements derived from the scenario in Section 3.2 related to the examples of security needs Ex1-Ex6. In the commitments below, debtor and creditor are roles, whereas antecedent and consequent are propositions.

Ex1. The non-repudiation security need results in a commitment from the secretary (sec) to the program coordinator (pc) that, if goal "write new document" is delegated to her, she will not repudiate the delegation:
C(sec, pc, d_1 =delegate(pc,sec,writeDocument), non-repudiation(d_1))

Ex2. The redundancy security need implies a commitment from the IS manager ism to the secretary that, if goal "obtain income statement" is delegated from sec to ism, goal redundancy will be guaranteed:
C(ism, sec, delegate(sec,ism,obtainIncomeStmt), redundancy(obtainIncomeStmt))

Ex3. The no-delegation security need for getting student personal data results in a commitment from ism to sec that, if the goal is delegated to sec, she will not further delegate the goal.
C(ism, sec, delegate(sec,ism,getStudentData), no-delegation(getStudentData))

Ex4. Since our modelling language does not make assumptions on the timing of authorizations, the non-disclosure security need of ism results in an unconditional commitment by sec for the non-disclosure of personal data and financial status. This means that, at any moment the secretary is in possession of those resources, she commits to not disclose them.
C(sec, ism, $\top$, non-disclosure(personalData $\wedge$ financialStatus))

Ex5. The need-to-know security need requested by the student (stud) for his personal data and financial status results in an unconditional commitment by ism that those resources will be used only in the scope of writing a document for the immigration office. Granted permissions are to produce and distribute tangible resources that represent those intangible resources.
C(ism, stud, $\top$, ntk(personalData $\wedge$ financialStatus, writeDocumentForIO, p $\wedge$ d)

Ex6. The integrity security need expressed by ism results in an unconditional commitment from sec to ism that personal data and financial status will not be modified.
C(sec, ism, $\top$, integrity(personalData $\wedge$ financialStatus))

4 Ongoing and future work

We are currently working on SecCo, which will be the socio-technical security modelling language for the EU-funded Aniketos project. Aniketos is about ensuring trustworthiness and security in composite services. Some topics we will investigate are:

- *Obligations view*: laws and organisational rules impose constraints on the way data is exchanged, stored, and used. This implies security needs that are not expressed by stakeholders, but that the service under design shall preserve.
- *Security needs*: we intend to significantly increase the number of security needs our language supports. Some examples are least privilege, anonymity, auditability, and written consent.
- *Formalization and reasoning*: we need to formally represent the models in the business view so to support automated reasoning to verify consistency (e.g. unauthorised resource provisions or delegations) as well as to derive security requirements for the composite service (the commitments that have to be preserved).
- *Deriving service interface specifications languages*: transforming the security requirements expressed as commitments into existing service interface specification languages. Our choice will depend on both language expressiveness and the existence of monitoring frameworks able to check service compliance with its interface.
- *Methodology and tool support*: we will devise a companion methodology in order to make the language applicable, and will build a full-fledged development tool based on the Eclipse Rich Client Platform.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grants no 257930 (Aniketos) and no 256980 (NESSOS).

References

1. Liu, L., Yu, E., Mylopoulos, J.: 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) 151–161
2. Giorgini, P., Massacci, F., Mylopoulos, J.: 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). Volume 2813 of LNCS., Springer (2003) 263–276
3. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-Oriented Extension of the Tropos methodology. International Journal of Software Engineering and Knowledge Engineering **17**(2) (2007) 285–309
4. Chopra, A.K., Dalpiaz, F., Giorgini, P., Mylopoulos, J.: Modeling and Reasoning about Service-Oriented Applications via Goals and Commitments. In: Proceedings of 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10). Volume 6051 of LNCS., Springer (2010) 113–128
5. Singh, M.P.: An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts. Artificial Intelligence and Law **7**(1) (1999) 97–113
6. Yu, E.S.K.: Modelling Strategic Relationships for Process Reengineering. PhD thesis, University of Toronto, Toronto, Ont., Canada, Canada (1996)
7. Dalpiaz, F., Paja, E., Giorgini, P.: Security Requirements Engineering via Commitments. In: Proceedings of the 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST 2011). To appear.