STS-Tool: Using Commitments to Specify Socio-Technical Security Requirements

Elda Paja, Fabiano Dalpiaz, Mauro Poggianella, Pierluigi Roberti and Paolo Giorgini

Department of Information Engineering and Computer Science,
University of Trento, Italy
{paja,dalpiaz,poggianella,roberti,giorgini}@disi.unitn.it

Abstract. In this paper, we present STS-Tool, the modelling and analysis support tool for STS-ml, an actor- and goal-oriented security requirements modelling language for Socio-Technical Systems (STSs). STS-Tool allows designers to model a socio-technical system at a high-level of abstraction, while expressing constraints (security needs) over the interactions between the actors in the STS, and derive security requirements in terms of social commitments (promises with contractual validity) once the modelling is done.

1 Introduction

Socio-Technical Systems (STSs) are an interplay of social (human and organisations) and technical subsystems, which interact with one another to reach their objectives, making a STS a network of social relationships. Each subsystem is a participant of the STS, and interacts with others through *message exchange*. But, participants in STSs are autonomous, heterogeneous and weakly controllable. This raises up a number of security issues when they interact, especially when interaction involves information exchange, since one might want to constrain the way information is to be manipulated by others. To deal with social aspects of the security problem in STSs, we have proposed to use *social commitments* [4] to constrain interaction. *Social commitments* are a purely social abstraction used to model interaction. They exist as a result of interaction: they are created and evolve while agents exchange messages.

The focus of our work is on security requirements engineering (SRE) for STSs, while allowing interacting parties to express their *needs* regarding security. We have proposed STS-ml [1] (Socio-Technical Security modelling language), an actor- and goal-oriented security requirements modelling language for STSs, to use our idea of relating security requirements to interaction. The language allows actors to express *security needs* over interactions to constrain the way interaction is to take place, and uses the concept of *social commitment* among actors to specify security requirements.

The notion of *social commitments* was first proposed by Singh [4], and we specialise it for the first time to represent security requirements. Other approaches to SRE either rely on high-level abstractions, such as goals or softgoals [3], or on technical mechanisms such as monitoring [2]. Instead, we concentrate on securing the interaction between actors. An important feature of *social commitments* that makes them adequate for this purpose, is that they have *contractual validity*. That is, non satisfaction of a *social commitment* might lead to further commitments to be made by the violator. In STS-ml they are used as a guarantee for the satisfaction of *security needs*: a commitment is made by an actor (*responsible*) to another actor (*requestor*) for the satisfaction of a *security need*. For instance, in e-commerce, a buyer (*requestor*) might want a seller not to disclose its credit card details to other parties, and to use this information strictly to perform the payment of the acquired goods. Once the buyer expresses these needs, the seller (*responsible*) is expected to commit to the buyer that credit card details will not be disclosed to other parties, and will be used only for the payment of the acquired goods. In this way, the list of *social commitments* is derived for each *security need* expressed by the stakeholders, and represents the security requirements specification for the system-at-hand. They prescribe the security properties stakeholders have to comply with in order for their interactions (and the STS) to be secure.

In this paper, we illustrate the usage of the concept of *social commitment* for the specification of security requirements. Specifically, we show how STS-Tool[1], the graphical modelling and analysis support tool for STS-ml, enables the derivation of security requirements expressed as *social commitments*.

2 Demonstration Content

Our demonstration will cover three main activities. *First*, we will show STS-Tool, the tool that supports modelling activities and the derivation of security requirements as proposed in STS-ml. STS-ml supports multi-view modelling: interactions among actors can be represented by focusing on orthogonal views. As shown in Fig. 1, STS-ml consists of three different views: *social*, *authorisation*, and *information*. The *security needs* are expressed in the *operational view* (Fig. 1), which consists of the three aforementioned views. The *operational view* is automatically mapped to the specification of *security requirements*, which supports the *security needs* expressed in the *operational view*. STS-Tool supports this feature, by providing different views on a diagram, showing specific elements while hiding others depending on the view one is working on. It performs *consistency checking* to help designers create diagrams that follow the semantics of STS-ml. Once the modelling is done, the tool offers designers the possibility to export the diagram (or the different views) to different file formats, such as png, etc.

Second, we will show the use of *social commitments* in serving as specification of security requirements for the system-to-be. For this purpose, we will show small examples to better explain how we capture interactions in STS-ml and how we derive the specification of security requirements.

Finally, we will show an already modelled scenario from a case study on e-Gov, developed as part of the European research project Aniketos[2]. The focus of this part of the demo will be on two aspects: *derivation of security requirements* and *generation of security requirements document*. For a more interactive demo, we will illustrate the features of STS-Tool by modelling a small scenario from the case study (*Example 1*).

[1] STS-Tool is available for download at http://fmsweng.disi.unitn.it/sts

[2] http://www.aniketos.eu/

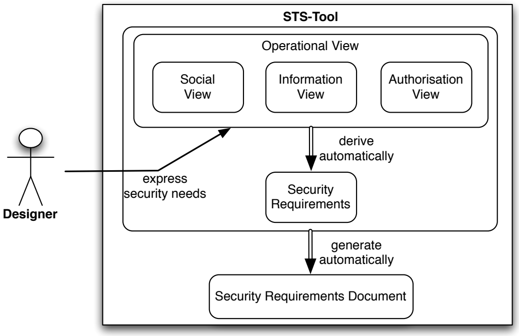


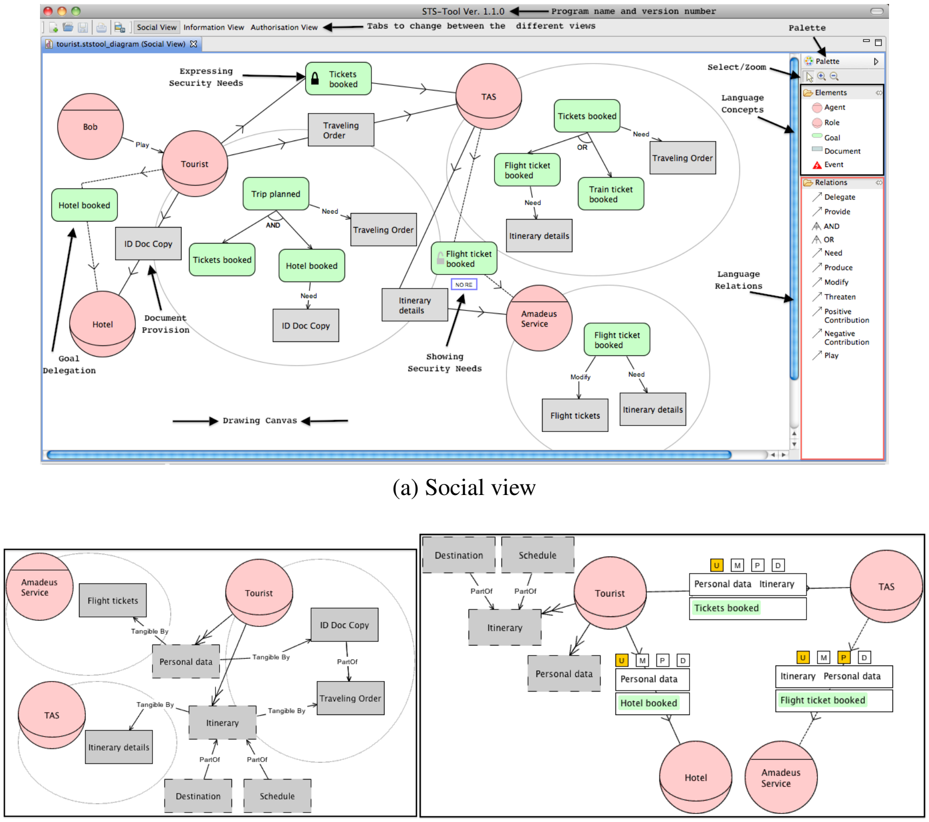
Fig. 1: From the operational view to security requirements

Example 1. Land selling involves not only finding a trustworthy buyer, but also exchanging several documents with various governmental bodies. The seller needs the municipality to certify that the land is residential zoning. The land selling process we consider is supported by an eGov application, through which the official contract (including the municipalitys certification) is sent to the ministry (who has the right to object) and is archived.

We will follow an iterative modelling process to model the different views: *social*, *information*, and *authorisation* view (Fig. 2) for the illustrating scenario. This will help us show how the tool facilitates and supports the modelling process.

Derivation of security requirements: we will show how the list of security requirements for the modelled scenario is derived once the modelling is done. STS-Tool allows the automatic derivation of security requirements, which are provided in a tabular form. The security requirements are listed, and they make clear the difference between actors that *request* a certain security need from those that are *responsible* for satisfying it. Security requirements can be sorted or filtered according to their different attributes. For instance, filtering the security requirements with respect to the *responsible* actor, gives an idea of who are the actors responsible to satisfy the commitments. On the other hand, filtering security requirements according to the *requirement type*, groups together commitments that need to be satisfied to fulfil a certain security need.

Generation of security requirements document: at the end of the modelling process, the tool allows designers to export models and generate automatically a *security requirements document*, which helps them communicate with stakeholders (Fig. 1). This document is customisable: designers can choose among a number of model features to include in the report (e.g., including only a subset of the actors, concepts or relations he wants more information about). However, the overall document provides a description of STS-Tool and communicates security requirements by providing details of each STS-ml view, together with their elements. The diagrams are explained in detail providing textual and tabular description of the models. The document is organised in sections, which the designer can decide to include or not in the generated document.



(a) Social view

(b) Information view

(c) Authorisation view

Fig. 2: Multi-view modelling for the eGov scenario

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

References

1. Fabiano Dalpiaz, Elda Paja, and Paolo Giorgini. Security requirements engineering via commitments. In *Proceedings of the First Workshop on Socio-Technical Aspects in Security and Trust (STAST'11)*, pages 1–8, 2011.
2. Paolo Giorgini, Fabio Massacci, John Mylopoulos, and Nicola Zannone. Modeling security requirements through ownership, permission and delegation. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, pages 167–176, 2005.
3. Lin Liu, Eric Yu, and John Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting. In *Proceedings of the 11th IEEE International Conference on Requirements Engineering (RE 2003)*, pages 151–161. IEEE Computer Society, 2003.
4. Munindar P. Singh. An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.