ST-Tool: A CASE Tool for Security Requirements Engineering[*]

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

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

John Mylopoulos
University of Toronto
jm@cs.toronto.edu

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

Abstract

Security Requirements Engineering is emerging as a branch of Software Engineering, spurred by the realization that security must be dealt with early on during the requirements phase. We propose ST-Tool, a CASE tool developed for modeling and analyzing functional and security requirements.

1 Introduction

Software designers have recognized the need to integrate most non-functional requirements (such as reliability and performance) into the software development processes, but security still remains an afterthought. This often means that security mechanisms have to be fitted into a pre-existing design which may not be able to accommodate them due to potential conflicts with functional requirements or usability.

This has spurred a number of researchers to model security requirements into "standard" software engineering methodologies. The major limitation of many proposals is that they treat security in system-oriented terms. In other words, they are targeted to *model a computer system* and the policies and access control mechanisms it supports. In contrast, to understand the problem of security engineering we need to *model the organization* and the relationships between all involved actors.

This paper presents ST-Tool, a CASE tool for design and verification of functional and security requirements. It has been designed to support the Secure Tropos methodology [3]. Main goals of the tool are:

- Graphical environment: a visual framework to draw functional and security requirements;
- Formalization: support to translate models into formal specifications;
- Analysis capability: a front-end to external tools for formal analysis.

2 Background

Secure Tropos [3] is an agent-oriented software development methodology, tailored to describe both the organization and the system with respect to functional and security requirements. Secure Tropos extends the Tropos methodology [1] and has the concepts of actor, service (i.e. goal, task, resource) and social relationships for defining the obligations among actors. A description of these concepts is provided in [3].

Various activities contribute to the acquisition of a first requirement model:

Actor modeling, which consists of identifying and analyzing both environment and system's actors.

Trust modeling, which consists of identifying actors which trust other actors for services, and actors which own services.

Delegation modeling, which consists of identifying actors which delegate to other actors the permission and/or execution on services.

Once the stakeholders have been identified, along with their goals and social relations, the analysis proceeds in order to enrich the model with further details. Goal refinement rests on the analysis of actor goals and is conducted by using AND/OR decomposition.

Due to lack of space, we have focused on the key modeling aspects of the framework and refer to [3] for the introduction of the formal framework based on Datalog.

3 Overview of ST-Tool

ST-Tool is mainly composed of two parts: ST-Tool kernel and external solvers. ST-Tool kernel has an architecture comprised of three major parts, each of which is comprised

[*]This work has been partially funded by the IST programme of the EU Commission, through an FET under the IST-2001-37004 WASP project, by the FIRB programme of MIUR under the RBNE0195K5 ASTRO project, also by the MOSTRO project funded by the Provincial Authority of Trentino. We thank Alberto Siena for supporting the development of ST-Tool.

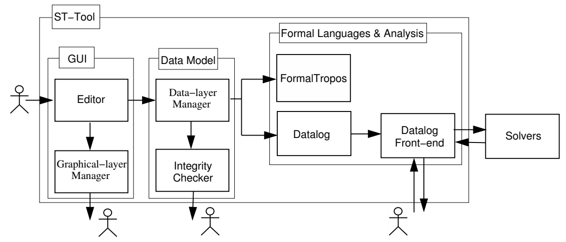
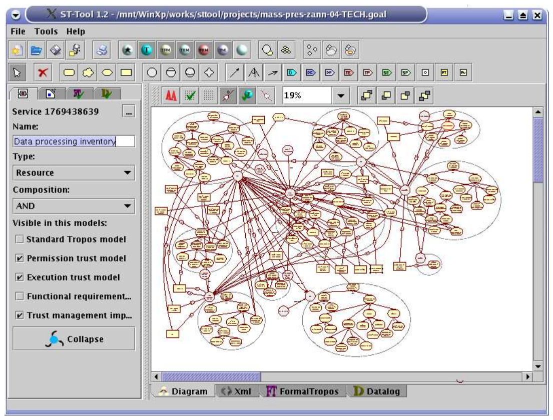


Figure 1. The Architecture Overview


ST-Tool 1.2 - /mnt/WinXp/works/sttool/projects/mass-pres-zann-04-TECH.goal
File
Tools
Help
19%
Collapse
Diagram
Xml
FormalTropos
Datalog

Service 1769438639
Name: Data processing inventory
Type: Resource
Composition: AND
Visible in this models:
- [ ] Standard Tropos model
- [x] Permission trust model
- [x] Execution trust model
- [ ] Functional requirement...
- [x] Trust management imp...

Figure 2. ST-Tool

of modules. Next, we discuss these modules and their interconnections. In Fig. 1, the modules of ST-Tool are shown, their interrelations are also indicated.

ST-Tool provides a graphical user interface (GUI), through which all its components are managed. A screen shot is shown in Fig. 2. The GUI's key component is the *Editor Module*. This allows designers to edit Secure Tropos models as graphs where nodes are actors and services, and arcs are relationships. A second component is the *Graphical-layer Manager (GM) Module* that aims to manage graphical objects. It supports goal refinement by associating a goal diagram with each actor. Further, GM permits to display one or more views of a diagram at the same time (dependency model, delegation model, trust model).

The *Data-layer Manager (DM) Module* is responsible for maintaining data corresponding to graphical objects. Its main task is to manage misalignments between relationships and their graphical representation. A support for detecting errors and warnings during the modeling phase is provided by the *Integrity Checker Module*. Integrity Checker reports errors such as "orphan relations" (i.e. incomplete relations) and "isolated nodes" (i.e. services not involved in any relations). Warnings are different from errors since designers may be perfectly happy with a design that does not satisfy them. Integrity Checker reports warnings when more than one service have the same name.[1]

[1]More than one service with the same name are needed to model delegation and trust chains.

After drawing so many nice diagrams, designers may want to check whether the model satisfies some general desirable properties. The tool allows an automatic transformation from Secure Tropos graphical models into Datalog and Formal Tropos [2] specifications. These are performed by two different modules: the *Formal Tropos Module* and the *Datalog Module*. The resulting specifications are displayed by selecting the corresponding panel. Since the formal semantics of Secure Tropos is based on Datalog, we mainly focus on Datalog Module. The intuitive descriptions of systems are often incomplete, and need to be completed for a correct analysis [3]. The *Datalog Front-end (DF) Module* provides support for model completing and checking by using external Datalog solvers. Essentially, DF permits designers to select properties to be verified and to specify additional security policies. Once designers are confident with the model, the resulting Datalog specification is verified by Datalog solvers with respect to the properties that designers want to check. Then, the solver output is parsed by DF in order to present it in a more user-readable format.

4 Conclusion

We have already used the tool to model a comprehensive case study on the compliance to the Italian legislation on Privacy and Data Protection by the University of Trento, leading to the definition and analysis of an ISO-17799-like security management scheme [4].

Future work will involve a front-end with T-Tool [2] for automatically verifying Formal Tropos specification. Further, Secure Tropos is still under work, so is ST-Tool, too. We are also considering to integrate our tools into the ECLIPSE platform.

References

[1] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. TROPOS: An Agent-Oriented Software Development Methodology. *JAAMAS*, 8(3):203–236, 2004.

[2] A. Fuxman, L. Liu, M. Pistore, M. Roveri, and J. Mylopoulos. Specifying and analyzing early requirements: Some experimental results. In *Proc. of RE'03*, page 105. IEEE Press, 2003.

[3] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Requirements Engineering meets Trust Management: Model, Methodology, and Reasoning. In *Proc. of iTrust'04*, *LNCS 2995*, pages 176–190. Springer-Verlag, 2004.

[4] F. Massacci, M. Prest, and N. Zannone. Using a Security Requirements Engineering Methodology in Practice: The compliance with the Italian Data Protection Legislation. *Comp. Standards & Interfaces*, 27(5):445–455, 2005. An extended version is available as Technical report DIT-04-103 at `eprints.biblio.unitn.it`.