Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Institut für Praktische Informatik und Medieninformatik
Fachgebiet für Verteilte Systeme und Betriebssysteme

Masters thesis

Graphical Specification Language for the Entity-Labeling Aspect

Submitted by:

Philipp Schwetschenau

Supervisor:	Prof. Dr.-Ing. habil. Winfried E. Kühnhauser
Supervisor:	Peter Amthor
Studies:	Computer science
Matriculation no.:	46756
Submission date:	Ilmenau, 29. November 2018

Contents

List of Figures

CHAPTER 1
Introduction

1.1 Domain

With the increasing number of IT systems, securing these systems became an obvious and important issue. For this purpose many security models and model families were developed for a wide field of application domains over the last years.

Formal security models offer possibilities to analyze them concerning security properties. However, because quantity and variety of these models grow just as much as their relevance for security-critical applications the model-based security engineering process became more complex and therefore error-prone to human deviations.

Amthor [2018] proposed a new approach called Aspect-oriented Security Engineering (AOSE) which claims to close semantic gaps between steps in the security engineering process (requirements, informal policy, formal model) to reduce the potential impact of human errors. This approach roughly adopts the idea of the aspect-oriented programming paradigm. It tailors all steps to aspects, which are non-functional requirements of the engineering process like determining requirements of policy semantics or analyzing certain security goals. There are two major classes for aspects regarding AOSE: related to policy semantics and to policy analysis.

One possible aspect of the former is the Entity Labeling Aspect (EL). It is designed to formally specify policy semantics typically found in operating systems and middleware systems. Therefore it bridges the gap and supports the transformation between informal policy and formal model. EL classifies model components into six semantic categories. The notation is on a mathematical basis and uses concepts like sets, assignments or constraints.

1.2 Motivation

The goal of reducing the impact of human errors by closing semantic gaps as much as possible with the help of AOSE/EL requires handling another formal notation, which is, in this case, EL itself and its classification into the six semantic categories. To support working with this approach, especially for the communication between different groups of people involved like model engineers, security architects, software developers or future administrators, who have to cooperate and coordinate and all have different levels of experience, Amthor [2018] considers a graphical representation to be helpful to enhance the transition between informal and formalized notation.

Amthor [2018] already uses visual representations to illustrate examples. However, the representations are not described in detail as well as they are inconsistent and ambiguous regarding several parts of the formalization (e.g. arrows can have several meanings and there is no way to specify functions with multiple input parameters). So these visualizations are appropriate to underline and support the engineering of an already formalized policy after EL-based model engineering, but are not suitable for independent modeling on a stand-alone level, because they are not equivalent to their formalized mathematical counterpart.

There are two types of visualization Amthor [2018] uses, which have different levels of abstraction: One is based on the actual model components and visualizes them and their relationships. The other one is based on a higher level of abstraction and visualizes the EL with its structure of semantic categories and their relationships.

There are other visual notations like UML or ERD, which have in common that they are tailored to particular needs in special application domains, so they can not be applied here, but may give inspiration on how to model certain semantics and relationships.

Eventually to be able to independently and visually model and work on EL-based policies there is need for an unambiguous formal graphical specification language.

1.3 Goal

The main goal is to develop a graphical specification language for EL-based security polices. This should focus on clean and unambiguous semantics of the language.

The use of the language should be as simple as possible and as comprehensive as needed. Therefore its appearance and elements should be clear and well-structured. A selection of appropriate symbols and geometrical shapes for their equivalent model counterparts has to be made regarding an intuitive understanding and workflow. This selection should not be designed contradictory or conflicting to already established notations, especially those, which may be used in the context of security engineering.

It should be evaluated how feasible and reasonable it is to develop equivalent counterparts for every possible element and relationship in context of EL and to display all of them at once. This may also lead to the question how the visualization is related to a visualization on a higher abstraction level and how they are connected and might be managed. The latter may be investigated as an optional goal.

In addition to that a GUI-based editor should be developed as a prototype to make use of the proposed graphical specification language.

1.4 Structure

CHAPTER 2 Fundamentals

This chapter provides information to all fundamentals necessary for this thesis. It will serve as basis for all design decisions in the following chapter 3 and chapter 4.

Section 2.1 introduces the security models HRU and RBAC as well as the Security Model Core. Section 2.2 focuses on the Aspect-oriented Security Engineering (AOSE) proposed by Amthor [2018]. It covers the Entity-Labeling Aspect in particular, one aspect of AOSE, which is going to be essential regarding the task to design a graphical specification language for it in chapter 3. In section 2.3 the well-established graphical notations UML and ER are described.

Based on human optical perception and gestalt laws section 2.4 provides information on how we perceive and evaluate visual impressions regarding two-dimensional forms and structures. In the last section 2.5 information on how to design a software graphical user interface are given with respect to common best practices like design patterns and modern usability.

2.1 Security models

Security policies are sets of rules to fulfill security related requirements of a system. To actually be able to work with such policies and also in context of analyzing them, they have to be formalized as an instance of a security model. In context of model-based security engineering these security models are obviously the most important artifact. So this section introduces the HRU security model as well as the family of RBAC models with its concepts and elements. Furthermore this section will present a concept to express all types of security models in form of a single model, the Security Model Core.

2.1.1 HRU

The HRU security model [Harrison, Ruzzo, and Ullman, 1975] is a fundamental, dynamic access control model. It manages permissions of subjects on objects with the help of an state automaton and access control matrices. While subjects are active (e.g. users), objects are passive (e.g. files).

Definition 2.1 (HRU).
A HRU model is a deterministic automaton (Q, Σ, δ, q_0) with $Q = 2^S \times 2^O \times M$ as state space, $M = \{m | m : S \times O \to 2^R\}$ as set of access control matrices, Σ as input alphabet, $\delta : Q \times \Sigma \to Q$ as state transition function and $q_0 \in Q$ as initial state.

2.1.2 RBAC

Role-based Access Control

Definition 2.2 (RBAC).
A RBAC$_0$ model is a tuple $(U, R, P, S, UA, PA, \mathit{user}, \mathit{roles})$ with U as set of *users*, R as set of *roles*, $P = 2^{O \times OP}$ as set of *permissions* (O is the set of *objects* and OP the set of *operations*), S as set of *sessions*, $UA \subseteq U \times R$ as *user-role relation*, $PA \subseteq P \times R$ as *permission-role relation*, $\mathit{user} : S \to U$ as assignment, that maps every session to a user and $\mathit{roles} : S \to 2^R$ as assignment, that maps every session to set of roles.

2.1.3 Security Model Core

2.2 Aspect-oriented Security Engineering and Entity-Labeling Aspect

2.2.1 Aspect-oriented Security Engineering

2.2.2 Entity-Labeling Aspect

2.3 Graphical notation models

2.3.1 Unified Modeling Language

The Unified Modeling Language (UML)

2.3.2 Entity-Relationship

Entity-Relationship (ER)

2.3.3 RBAC notation by Sandhu

Sandhu useda notation to visualize his proposal to roled-based access control

maybe my own gsl from bachelor thesis?

2.4 Gestalt laws and human optical perception

2.4.1 Gestalt laws

2.4.2 Human optical perception

2.5 GUI Design

2.5.1 Design Patterns

2.5.2 Usability

CHAPTER 3
Design: Graphical specification language

3.1 Concept

approach, basic ideas, adoptions from literature

3.2 Elements

3.3 Relationships

3.4 Structure

3.5 Visualization on the higher abstraction level

CHAPTER 4
Design: Editor GUI

4.1 Structure

4.2 Sections

CHAPTER 5
Implementation

5.1 Implementation base)

Qt, MVC

5.2 Structure

5.3 GUI sections

CHAPTER 6
Evaluation

6.1 Graphical specification language

6.2 Editor

CHAPTER 7
Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

CHAPTER 8
Summary

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Bibliography

Peter Amthor. *An Aspect-oriented Approach to Model-based Security Engineering.* dissertation, Technische Universität Ilmenau, 2018.

Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. On Protection in Operating Systems. *Operating Systems Review, special issue for the 5th Symposium on Operating Systems Principles*, November 1975.