Graphical Specification Language for the Entity-Labeling Aspect

Exposé – Master Thesis

Philipp Schwetschenau
Technische Universität Ilmenau
philipp.schwetschenau@tu-ilmenau.de

May 17, 2018

1 Introduction

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.

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.

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.

4 Approach

Based on cognitive psychology like gestalt laws and human optical perception on the one hand and based on usability and common best practices on the other hand, a graphical specification language should be developed. For this already existing and established visual notations like UML or ERD should be analyzed regarding their way of visualizing structures and relationships for similar problems. Especially the visual notation used by Sandhu [1996] should be investigated due to its tailoring to the domain of security models and its relevance in literature, e.g. in the NIST RBAC model. This may lead to adopting ideas from those visual notations or at least get inspired by them.

A starting point could be the study and evaluation of the visual notation used by Amthor [2018]. This should give hints on what might be missing, what might be ambiguous and what are potential requirements for a new graphical specification language.

The development as well as the evaluation of the language should be supported by examples with different scope and complexity.

References

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

Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, Charles E. Youman, *Role-Based Access Control Models*, IEEE Computer, 29(2):38–47, February 1996