**Faculdade de Engenharia da Universidade do Porto**



**Physical Access Control System**

**Mestrado Integrado em Engenharia Informática e Computação**

**Métodos Formais em Engenharia de Software**

Bruno Moreira

Márcio Fontes

November, 2015

**Abstract**

On this report we present a formal, tool-supported approach to the design and maintenance of access control policies expressed in the eXtensible Access Control Markup Language (XACML). Our aim is to develop an application using the model-oriented specification language from Vienna Development Method (VDM++), capable of perform actions based on targets, subjects and subjacent policies, and therefore apply the specified policy combination algorithms to determine its outcome status (e.g., denial, permit, etc.).

Content

[1. Introduction 4](#_Toc436671640)

[1.1 Project Description 4](#_Toc436671641)

[1.2 Objectives 4](#_Toc436671642)

[1.3 Requirements 5](#_Toc436671643)

[2. UML Modeling 6](#_Toc436671644)

[2.1 Use Case Diagram 6](#_Toc436671645)

[2.2 Class Diagram 6](#_Toc436671646)

[3. VDM++ Modeling 7](#_Toc436671647)

[3.1 Classes 7](#_Toc436671648)

[3.2 Data Types 7](#_Toc436671649)

[3.3 Domains 7](#_Toc436671650)

[4. Model Validation 8](#_Toc436671651)

[4.1 Test Classes 8](#_Toc436671652)

[4.2 Test Results 8](#_Toc436671653)

[4.3 Requirements Traceability 8](#_Toc436671654)

[5. Model Verification 9](#_Toc436671655)

[5.1 Domain Verification 9](#_Toc436671656)

[5.2 Invariant Verification 9](#_Toc436671657)

[6. Code Generation 10](#_Toc436671658)

[7. Conclusions 11](#_Toc436671659)

[7.1 Results Achieved 11](#_Toc436671660)

[7.2 Improvements 11](#_Toc436671661)

[7.3 Effort 11](#_Toc436671662)

[References 12](#_Toc436671663)

# **1. Introduction**

## **1.1 Project Description**

This project aims to develop a physical access control system using XACML[[1]](#footnote-1) language, implemented in VDM++, in order to perform authorization, identification, authentication, access approval and keep records of all succeeded or failed access requests.

## **1.2 Objectives**

The physical access control system should have the following features:

* may be used in all sorts of physical facilities, such as hotels, schools, banks, military facilities, etc.;
* should be able to control the access to buildings, sectors (inside a building), rooms, parking lots, floors (in elevators), and other facilities;
* each authorized user is given a contactless card to present at appropriate access points, communicating with NFC (near field communication) or other means;
* access cards may be temporary, with a defined date-time of expiration (e.g., for hotel guests);
* each access card has a unique identifier and access cards may be reused;
* both users and facilities may be organized into groups (e.g., students, teachers, classrooms, computer laboratories, etc.) to facilitate the definition of access rules;
* a user or facility may belong to multiple groups;
* access policies are defined by means of access rules;
* each access rule specifies a user or group of users, a facility or group of facilities, and possibly a temporal constraint (a specific date-time interval, a recurrent time interval, etc.);
* rules may be defined as exceptions to other rules (e.g., to deny access for some period of time);
* the system should be able to decide on access requests;
* the system should keep a log of all succeeded or failed access requests.

## **1.3 Requirements**

This project was implemented based on the following requirements:

* **R1**: Provide a method for combining individual **rules** and **policies** into a single **policy set** that applies to a particular decision **request**;
* **R2**: Provide a method for rapidly identifying the **policy** that applies to a given **action**, based upon the values of **attributes** of the **subjects**, **resource** and **action**;
* **R3**: Provide a method for basing an **authorization decision** on the contents of an information **resource**;
* **R4**: Provide a method for flexible definition of the procedure by which **rules** and **policies** are combined;
* **R5**: Provide a method for specifying a set of **actions** that must be performed in conjunction with **policy** enforcement;

# **2. UML Modeling**

On this section it’s presented the use cases and conceptual model for this project, as well as additional notes and constraints concerning the diagrams.

## **2.1 Use Case Diagram**

## 

## **2.2 Class Diagram**

Conceptual modelling is the abstraction of a simulation model from the part of the real world it is representing - “the real system” (Robinson, 2008). After collecting the necessary requirements, we achieved the following conceptual model, represented by (Figure 1):

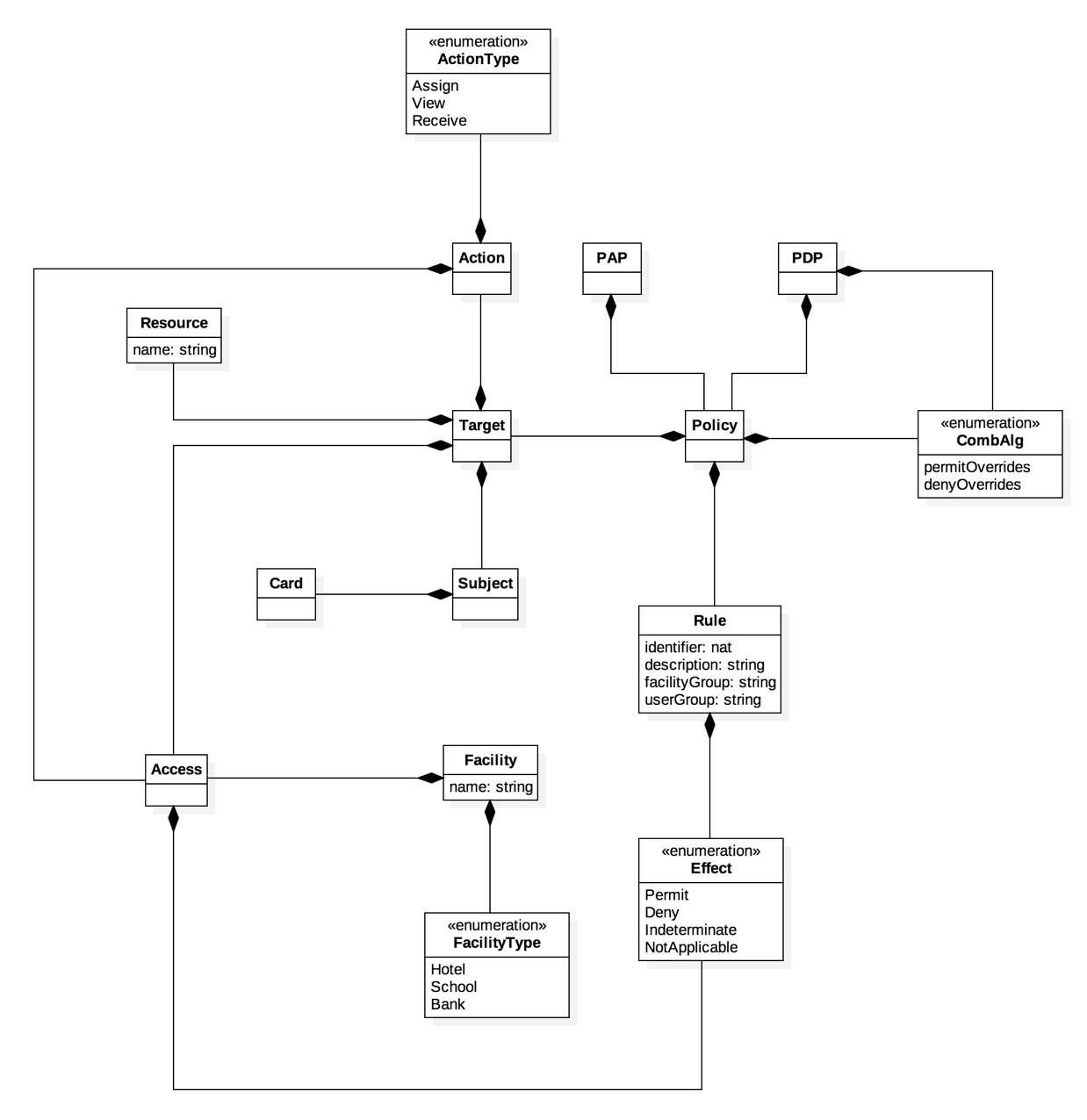


Figure 1 - UML Class Diagram

# **3. VDM++ Modeling**

## **3.1 Classes**

## **3.2 Data Types**

## **3.3 Domains**

# **4. Model Validation**

## **4.1 Test Classes**

## **4.2 Test Results**

## **4.3 Requirements Traceability**

# **5. Model Verification**

## **5.1 Domain Verification**

## **5.2 Invariant Verification**

# **6. Code Generation**

# **7. Conclusions**

## **7.1 Results Achieved**

## **7.2 Difficulties**

We honestly think that there should be more emphasis on explaining how using VDM++ may benefit the way programmers develop applications, using imperative languages like Java. There’s been a certain difficulty at the beginning to actually know what to do and where to start, and we lost tons of time on that dilemma. Furthermore, the massive amount of information about XACML was quite misleading at the beginning since we had no idea if we should implement XACML in its total completeness, as there wouldn’t be enough time to develop a system with that kind of scope. This led to delays on the development and therefore the application’s quality was far from we expected.

## **7.3 Improvements**

After developing this (quite) simple physical access control system using XACML, there are some features which we would like to implement, and therefore take use of all VDM++ potential capabilities. The first enhancement would be to translate the user request into a XACML type-request in order to follow the control flow in XACML. The second enhancement would be to read a XML file containing the policies, already in XACML, and populate the set of policies. The last enhancement would be to export a file with all the requests, taken actions and combining algorithms used.

## **7.4 Effort**

The distribution of effort (%) by each group member is given as follows:

* Bruno Moreira –
* Márcio Fontes –

# **References**

Bryans, J. W., & Fitzgerald, J. S. Formal Engineering of XACML Access Control Policies in VDM++. Newcastle University, School of Computer Science. Newcastle: Newcastle University.

OASIS. (2013 de january de 2013). eXtensible Access Control Markup Language (XACML) Version 3.0. Acessed on December 2nd, 2015, available at OASIS Docs: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

Robinson, S. (2008). Conceptual Modelling for Simulation Part I: Definition and Requirements. Journal of the Operational Research Society.

1. XACML – eXtensible Access Control Markup Language [↑](#footnote-ref-1)