UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

An aspect-oriented approach towards enhancing Optimistic Access Control with Usage Control

by

Keshnee Padayachee

submitted in fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in the subject of

COMPUTER SCIENCE

in the

Faculty of Engineering, Built Environment and Information Technology

at the

UNIVERSITY OF PRETORIA

SUPERVISOR: Prof. J.H.P. Eloff

DECEMBER 2009

PREFACE

This research was conducted on a part-time basis between 2004 and 2009 in collaboration with the Department of Computer Science at the University of Pretoria under the supervision of Professor J.H.P. Eloff. The results are the original work of the author and have not been submitted for any degree at any other tertiary institution.

ABSTRACT

With the advent of agile programming, lightweight software processes are being favoured over the highly formalised approaches of the 80s and 90s, where the emphasis is on "people, not processes". Likewise, access control may benefit from a less prescriptive approach and an increasing reliance on users to behave ethically. These ideals correlate with optimistic access controls. However, such controls alone may not be adequate as they are retrospective rather proactive. Optimistic access controls may benefit from the stricter enforcement offered by usage control. The latter enables finer-grained control over the usage of digital objects than do traditional access control policies and models, as trust management concerns are also taken into consideration. This thesis investigates the possibility of enhancing optimistic access controls with usage control to ensure that users conduct themselves in a trustworthy manner. Since this kind of approach towards access control has limited applicability, the present study investigates contextualising this approach within a mixed-initiative access control framework. A mixed-initiative access control framework involves combining a minimum of two access control models where the request to information is mediated by a mixture of access policy enforcement agents. In order for this type of integration to be successful, a software development approach was considered that allows for the seamless augmentation of traditional access control with optimistic access control enhanced with usage control, namely the aspect-oriented approach. The aspect-oriented paradigm can facilitate the implementation of additional security features to legacy systems without modifying existing code. This study therefore evaluates the aspect-oriented approach in terms of implementing security concerns.

It is evidently difficult to implement access control and in dynamic environments preconfigured access control policies may often change dramatically, depending on the context. In unpredicted circumstances, users who are denied access could often have prevented a catastrophe had they been allowed access. The costs of implementing and maintaining complex preconfigured access control policies sometimes far outweigh the benefits. Optimistic controls are retrospective and allow users to exceed their normal privileges. However, if a user accesses information unethically, the consequences could be

disastrous. Therefore it is proposed that optimistic access control be enhanced with some form of usage control, which may prevent the user from engaging in risky behaviour.

An initiative towards including security in the earlier phases of the software life cycle is gaining momentum, as it is much easier to design with security from the onset than to use the penetrate-and-patch approach. Unfortunately, incorporating security into software development takes time and developers tend to focus more on the features of the software application. The aspect-oriented paradigm can facilitate the implementation of additional security features in legacy systems without modifying existing code. The current study evaluates the aspect-oriented approach towards enhancing optimistic access control with usage control. The efficacy of the aspect-oriented paradigm has been well established within several areas of software security, as aspect-orientation facilitates the abstraction of these security-related tasks so as to reduce code complexity.

SUMMARY

Title: An aspect-oriented approach towards enhancing Optimistic Access Control with Usage Control

Candidate: Keshnee Padayachee

Supervisor: J.H.P. Eloff

Department: Department of Computer Science, Faculty of Engineering, Built Environment and Information Technology

Degree: Doctor of Philosophy in Computer Science

Keywords: Usage Control, Optimistic Access Control, Access Control, Aspect-Oriented Programming

ACKNOWLEDGEMENTS

This work would not have been possible without the support and encouragement of my supervisor Professor J.H.P. Eloff.

I am grateful to my husband Devern Padayachee; my nephews Ryan and André Veerasamy; and my colleagues at UNISA, especially Professor Elmé Smith for giving me the requisite courage to prevail. I also give credit to providence for giving me the perseverance to continue in spite of the odds.

It always seems impossible until is done
-Nelson Mandela

TABLE OF CONTENTS

PART 1: 1

CHAPTER 1: INTRODUCTION 2

1.1 Introduction 2

1.2 Motivation for this study 5

1.3 Problem Statement 9

1.4 Terminology used in this thesis 9

1.5 Research Methodology 10

1.6 Delimitations 11

1.7 Thesis Layout 11

1.8 Summary 14

CHAPTER 2: ACCESS CONTROL 15

2.1 Introduction 15

2.2 Discretionary Access Control 15

2.3 Mandatory Access Control 17

2.4 Role-based Access Control 19

2.5 Conclusion 20

CHAPTER 3: OPTIMISTIC ACCESS CONTROL 22

3.1 Introduction 22

3.2 Optimistic Access Control 23

3.3 Requirements for Optimistic Security 24

3.4 Applicability of optimistic security 25

3.5 The extensibility of the Optimistic Access Control Model 26

3.6 Conclusion 29

CHAPTER 4: USAGE CONTROL 30

4.1 Introduction 30

4.2 The continuity and mutability of the UCON model 31

4.3 The ABC Model for Usage Control (UCON model) 32

4.4 The Usage Control Model architecture 34

4.5 The Applicability and Extensibility of the UCON model 36

4.6 Conclusion 37

PART 2: 39

CHAPTER 5: ASPECT-ORIENTED PROGRAMMING 40

5.1 Introduction 40

5.2 Evolution to Aspect-Oriented Programming 41

5.3 Aspect-Oriented Programming Terminology 43

5.4 AOP Frameworks 44

5.5 Evaluating Aspect-Oriented Programming 46

5.6 Conclusion 48

CHAPTER 6: ASPECT-ORIENTED SECURITY 49

6.1 Introduction 49

6.2 Aspect-oriented programming and its application to security 50
6.2.1 Access Control and Authentication 50
6.2.2 Accountability and Audit 52
6.2.3 Cryptographic Controls 52
6.2.4 Information Flow Controls 53
6.2.5 Protection from invasive software 53
6.2.6 Security kernels 54
6.2.7 Verification 54

6.3 Conclusion 55

PART 3: 56

CHAPTER 7: THE OAC(UCON) MODEL 57

7.1 Introduction 57

7.2 A motivating example 57

7.3 Architecture 59

7.4 Detailed Design 64
7.4.1 Formal Specifications 64
7.4.2 The Use Case Diagram of Usage Control under the Optimistic Access Control Paradigm 67

7.5 Conclusion 69

CHAPTER 8: PROTOTYPING AND MODEL EVALUATION 70

8.1 Introduction 70

8.2 The aim of the proof-of-concept prototype 70

8.3 Implementation of the proof-of-concept prototype 71

8.4 An implementation overview of the proof-of-concept prototype 76

8.5 Proof-of-concept prototype operation 80

8.6 Evaluation of the Aspect-Oriented Approach 83
8.6.1 The Design Approach 83
8.6.2 Execution Time and Memory Usage 85

8.7 Evaluation of the model concept 87

8.8 Conclusion 93

CHAPTER 9: CONCLUSION 95

9.1 Introduction 95

9.2 Main contribution 95

9.3 Revisiting the problem statement 97

9.4 Future Research Directions 98

9.5 Conclusion 99

REFERENCES 100

INDEX 116

APPENDICES

Appendix A: List of Publications **118**

Appendix B: OOP Documentation **122**

Appendix C: AOP Documentation **138**

Appendix D: Prototype Evaluation **157**

Appendix E: Data Collection **165**

Appendix F: AspectJ Semantics **171**

Appendix G: Running the Demo Project **172**

LIST OF FIGURES

Figure 1.1: Overview of Thesis 13

Figure 2-1: Discretionary Access Control based on an Access Control List (adapted from(Tolone et al., 2005)) .. 16
Figure 2-2: Mandatory access control (MAC) (adapted from (Russell and Gangemi, 1991)) 17
Figure 2-3: Role-based Access Control (adapted from (Samarati and de Capitani di Vimercati, 2001)) 19

Figure 4- 1: Continuity and Mutability Properties(Park et al.). 31
Figure 4-2: ABC Model Components(Sandhu and Park, 2003) 32
Figure 4-3: Conceptual Structure for the UCON Reference Monitor (Sandhu and Park, 2003) 35

Figure 5-1: Illustration of the Weaving Concept 44

Figure 7-1: Architectural Diagram 59
Figure 7-2: Conceptual Structure for Optimistic Access Control enhanced with Usage Control 61
Figure 7-3: A Mixed-Intiative Access Control Framework – combining RBAC with OAC(UCON) 64
Figure 7-4:Use Case Diagram of OAC(UCON) 67

Figure 8-1: State Activity diagram of OAC(UCON) Model) 71
Figure 8-2: Thread Diagram of the OAC(UCON) model 72
Figure 8-3: UML Diagram showing Aspect UsageControlInjector and Core Classes 75
Figure 8-4: Showing the OOP UML of Core Classes 84
Figure 8-5: OOP package level diagram vs AOP package level diagram (on the right) 85
Figure 8-6: Showing comparisons of the execution time of OO vs AOP 86
Figure 8-7: Showing comparisons of and Memory Usage of OO vs AOP 86

PROGRAM LISTINGS

Program Listing 6-1: Generalised Aspect Code for Access Control *51*
Program Listing 6-2: Demonstrating Accountability and Auditing with Aspect-Orientation *52*

Program Listing 8-1: 'SampleAuthorization' class *73*
Program Listing 8-2: Showing the `UsageControlInjector` *Aspect* *77*
Program Listing 8-3: Depicting an InterTypeDeclaration Aspect *79*