DOCTORAL DISSERTATION

Access Control Management in Distributed Object Systems

GERALD BROSE

2001

submitted at the
Department of Mathematics and Computer Science
Freie Universität Berlin

Supervisors:
Prof. Dr. Klaus–Peter Löhr
Prof. Dr. Dieter Gollmann

Datum der Disputation: 17. Oktober 2001

Gerald Brose
Marzahnstraße 23
13509 Berlin
gerald.brose@acm.org

Contents

1 Introduction 11

1.1 Middleware and Security . . . 11

1.1.1 Security and Complexity in Distributed Systems . . . 12

1.1.2 Access Control . . . 13

1.2 Problem Statement . . . 14

1.3 Contributions . . . 16

1.4 Thesis Overview . . . 16

2 Requirements for Manageable Access Control 19

2.1 Policy–related Management Tasks . . . 19

2.1.1 Principals and Credentials Management . . . 20

2.1.2 Object and Domain Management . . . 23

2.1.3 Policy Management . . . 24

2.2 Policy Deployment . . . 27

2.3 Policy Issues in Application Development . . . 28

2.3.1 Requirements Analysis . . . 28

2.3.2 Design . . . 32

2.4 Summary of Requirements . . . 33

3 Standard CORBA Security 35

3.1 Overview of CORBA Security . . . 35

3.1.1 Principals and Authentication . . . 36

3.1.2 Secure Invocation . . . 37

3.1.3 Delegation and Secure Interoperability . . . 40

3.1.4 Non–Repudiation . . . 44

3.1.5 Security Policy Domains . . . 44

3.2 The default CORBA access model . . . 45

3.2.1 Principals and Attributes . . . 46

3.2.2 Rights . . . 46

3.2.3 An example policy . . . 47

3.3 Evaluation of Requirements 50
3.3.1 Principals and Roles 51
3.3.2 Management 52
3.3.3 Restrictions of the access model 52

4 View–Based Access Control 55
4.1 The View Policy Language 55
4.1.1 Policy 55
4.1.2 Roles 56
4.1.3 Views 57
4.1.4 Implicit Authorizations, Denials, and Conflict Resolution 60
4.1.5 Dynamic rights changes 64
4.1.6 Conditional and Virtual Views 68
4.2 A Discussion of the View–Based Access Model 69
4.2.1 Principals and Roles 69
4.2.2 A typed matrix model 71
4.2.3 Constraints 72
4.2.4 Views and Matrix Constraints 76
4.2.5 Explicit Discretionary Assignment and Removal 77
4.2.6 Implicit Assignment and Removal 79
4.2.7 Conditional and Virtual Views 85

5 Model Formalization 87
5.1 Protection State 87
5.2 Groups, Roles, and Accesses 88
5.3 Views 90
5.3.1 Configuration Constraints 90
5.3.2 Definition Constraints 91
5.3.3 View extension 92
5.3.4 Priorities and conflict resolution 92
5.4 Authorization Scheme 93
5.4.1 Creating and Deleting Principals 94
5.4.2 Creating and Deleting Objects 95
5.4.3 Creating and Deleting Matrix Entries 95
5.5 Access Decisions 99

6 An Application Case Study 101
6.1 Requirements Analysis 101

6.1.1 Functional Requirements 101
6.1.2 Application Security Requirements 103
6.2 Application design 103
6.3 Policy design 106
6.3.1 Roles, Constraints, and Initial Views 106
6.3.2 Dynamic aspects 108

7 An Infrastructure for View–Based Access Control 115
7.1 Language and Deployment Tools 115
7.2 Implementation of the Access Control Mechanism 117
7.2.1 Authentication 118
7.2.2 Schemas 120
7.3 Role Management 121
7.3.1 Role Interceptors 121
7.3.2 The Role Server and Repository 123
7.3.3 A Role Management Tool 124
7.4 Domain Management 126
7.4.1 A model of policy domains 126
7.4.2 Implementation Aspects 129
7.4.3 An Access Control Example 130
7.5 Policy Management 132
7.5.1 Policy Server and View Repository 132
7.5.2 A Policy Management Tool 133
7.6 Architecture Evaluation 134
7.6.1 CORBA Integration and Portability 135
7.6.2 Performance Implications 135
7.6.3 Architectural Complexity and Protection of the Infrastructure 137

8 Related Work 139
8.1 Earlier View Concepts 139
8.2 Programming Language Approaches to Protection 140
8.3 Policy Languages 143
8.4 Role–Based and Task–Based Models 146
8.5 Other Related Work 147

9 Summary and Conclusions 149
9.1 Summary 149
9.2 Evaluation 150

9.2.1 Application Areas 151
9.3 Future Work 151

A VPL Grammar and XML DTD 153
A.1 LALR Grammar 153
A.2 XML Document Type Definition 156

B IDL and VPL Definitions for the conference example 159
B.1 IDL 159
B.2 VPL 161

C Zusammenfassung in deutscher Sprache 167

D Lebenslauf 169

E Erklärung 171

Bibliography 173

Index 182

List of Figures

2.1 Steps in the Development of a Security Policy. 29
2.2 A Use Case diagram. 30

3.1 Credentials. 37
3.2 Secure Invocation. 38
3.3 ISO Access Control Model. 39
3.4 Delegation of Invocations. 40
3.5 Name server interface. 47
3.6 Name Service use cases. 48
3.7 Principals make statements. 51
3.8 Principals invoke operations. 52

4.1 An example policy. 56
4.2 Role declarations. 57
4.3 A view definition. 58
4.4 The Document interface. 58
4.5 View extension. 59
4.6 Denials and explicit priorities. 61
4.7 Potentially conflicting views. 62
4.8 Conflict between strong rights. 62
4.9 Types and the inheritance of operations. 63
4.10 An assignable view. 65
4.11 Interface DocumentFactory. 66
4.12 Views and schema for document creation. 66
4.13 Conditional access rights. 68
4.14 Groups and Roles. 71
4.15 Role declarations. 75
4.16 A conflict–free schema. 82
4.17 Potential conflicts in a schema. 82
4.18 Multiple Schemas. 83

4.19 Matrix restrictions. 86

6.1 Actors and Use Cases for the Conference System. 102
6.2 The ConferenceManagement interface. 104
6.3 The SubmissionManagement interface. 104
6.4 IDL Interfaces for the conference application. 105
6.5 Author and Reviewer tools. 105
6.6 Role declarations, constraints, and initial views. 106
6.7 Views. 107
6.8 The PaperReviewing view. 107
6.9 Views on documents and papers. 108
6.10 The Steering schema and virtual views. 109
6.11 The Submission schema and views for authoring. 110
6.12 The Paper schema (submission part). 110
6.13 The Paper schema (reviewing part). 111
6.14 Views for the reviewing phase. 112
6.15 The NoMorePaperReviewing view. 113

7.1 Policy development. 116
7.2 Policy deployment. 117
7.3 The DeploymentTarget interface. 117
7.4 Access Control with Interceptors. 118
7.5 Authentication and Interceptors. 119
7.6 A client–side role interceptor. 121
7.7 A server–side role interceptor. 122
7.8 The RoleServer interface. 123
7.9 Role server and repository. 124
7.10 Role Management GUI. 125
7.11 A group graph with role assignments. 126
7.12 A domain graph. 128
7.13 A domain graph in the Domain Browser. 131
7.14 A meta–policy. 131
7.15 Object associations. 133
7.16 The PolicyContainer interface. 134
7.17 The Policy Management GUI. 135

Abstract

The main question addressed in this work is how the specification, deployment and management of application–oriented access control policies in distributed object systems can be supported in a way that increases the overall security. The first chapters of this thesis examine the problems that need to be addressed and identify a number of requirements for manageable access control. The overall management task is analyzed and structured into subtasks that are performed by potentially separate managers: principals or credentials management, object and domain management, and policy management. Also, the tasks of policy deployment and development are examined. As a result, we identify the requirements for documentation, support for communication between the involved parties, and for suitable management abstractions. It is concluded that an integrated approach to secure software development and management is required and that it can best be supported by the definition of a declarative policy language. Looking at the current technology for CORBA security reveals conceptual scalability problems and lack of structured support for policy design.

Therefore, this thesis proposes a new view–based access model and a declarative specification language called *view policy language* (VPL). The abstractions of this language are designed to support deployment and development as well as management of application policies. The central concepts of VPL are views as a first–class concept for the type–safe aggregation of access rights, roles as a task–oriented abstraction of callers, and schemas as a means of specifying triggered dynamic changes in the protection state. To prove the practical relevance of these concepts, a comprehensive case study is analyzed and implemented. The technical feasibility of view–based access control is shown through an implementation of the required security infrastructure, which includes an interceptor–based access control mechanism, a language compiler, view and role repositories, and graphical management tools.