


Role Based Reengineering of Web Applications*

A. De Lucia, M. Giordano, G. Polese, G. Scanniello, G. Tortora
Dipartimento di Matematica e Informatica
Università di Salerno,Italia

* This research has been supported by “Centro Regionale di Competenza – ICT” of Regione Campania, Italy

Agenda

- Motivation
- Access policies
- Reengineering approach
- A case study
- Final remarks and future work

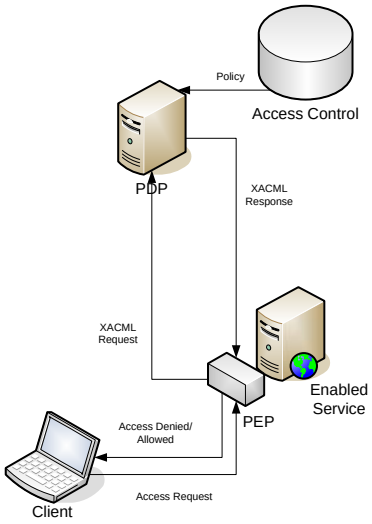
Motivation

- WAs manage a huge number of access rules and constraints for secure access to sensitive data and software resources
- The management of secure accesses might be simplified when based on policies
- Conceptualize the hardware characteristics providing powerful abstractions of systems
- Policies are defined by languages requiring specific knowledge by the administrator

Access policies

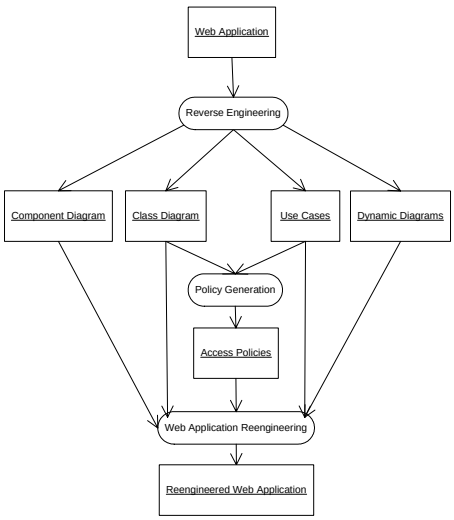
- Role Based Access Control (RBAC) model
- It is used in contexts where the privileges to use a resource are connected to a role and not to a specific user
- RBAC access policies are modelled by a suite visual languages
  - Role Diagram allows defining the roles and their hierarchic relations
  - Permission Diagram is proposed to define access permissions for a resource and to associate them to a role
  - Separation of Duty Diagram allows defining the mutually exclusive relations between users or subjects and a set of roles
  - Policy Assignment Diagram is proposed to define the association between a user or subject and roles
- A Java prototype turns the access policies into XACML (eXtensible Access Control Markup Language) format

Policy Management



- Policy Enforcement Point (PEP)
- Policy Decision Point (PDP)
- The PEP receives an access request from the application and codifies it into XACML
- The XACML request is sent to the PDP, which judges it
- The PEP receives an XACML response from the PDP
- The authorization decision allows the PEP to enforce the access

Reengineering approach



- Rounded rectangles represent process phases
- Rectangles represent the generated artefacts
- The three phases:
  - *Reverse Engineering*,
  - *Policy Generation*,
  - *Web Application Reengineering.*

Reverse Engineering (1 of 2)

- Actors' roles interacting with the applications and the software resources they access, including functionalities, classes, and components have to be identified
- Static and dynamic analyses on the legacy WA have to be performed
- The result of this phase is a set of diagrams:
  - class diagram
  - use case and sequence diagrams
  - component diagram
  - dynamic diagrams
- The first version of class and component diagrams is achieved by static analysis of the WA

Reverse Engineering (2 of 2)

- The class diagram is achieved from the analysis of both the WA code and the database schema
  - It is useful to identify the users of the application and their roles
- The component diagram is required to abstract the structure of the WA and identify the best way to reengineer it
- Both the diagrams are refined during dynamic analysis
- During the dynamic analysis
  - Sequence and use cases diagrams are produced to identify the interactions between users and WAs
  - Actions that the actors can perform on a given resource are identified to abstract them in secure access policies
  - To abstract business processes of the WA as UML activity diagrams to detect how users change their role during the use of the application

Policy Generation

- According to the RBAC model the access and control policies are specified considering the identified roles and resources
- Policies should describe the kind of users that can benefit from a given application, the priority in the using of it, and the network resources that can be granted to the users.
  - users,
  - resources,
  - applications,
  - security characteristics,
  - factory priorities,
  - network features.

Web Application Reengineering

- The application architecture in restructured and its pages are modified to manage the control and access policies
- Starting from the results of both the static and dynamic analyses, the software engineer decides where PEP components have to be placed in the restructured WA
- The component diagram of the resulting application is enhanced introducing PEP components
- Finally, the system database could also be restructured to enable the management of users and their roles

A Case Study

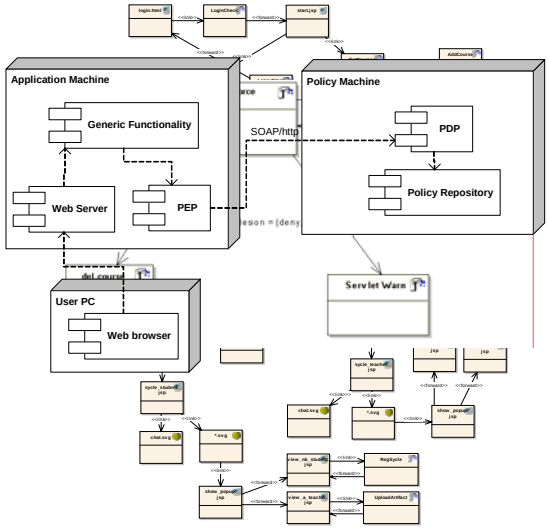
- The WA we used as case study is SYCLE (SYnchronous Collaborative Learning Environment) Server
- This application allows instructional designers and teachers to manage synchronous collaborative environments defined by the SYCLE editor
- Groups of students use this application to work together to produce artefacts and solve related problems
- SYCLE server has been developed using:
  - Apache,
  - Tomcat,
  - Java Server Pages,
  - Servlet,
  - Interbase and MySQL

SYCLE Server

- The original application consisted of 66 files
- The overall size of the application was 1.57 MB
- The application also contained images, stylesheet files, and XML files

Type	*Number*
Static pages	2
Dynamic pages	22
Servlet	18
Java Classes	24

Reengineering view



Final remarks and future work

- In this paper a role based approach to reengineer WA has been presented
- It starts from static and dynamic analysis to comprehend the WA, and looks for roles and the software resources
- The access policies in the reengineered WA are enforced by the PEP and PDP software components
- A preliminary case study has also been proposed
- Future work will be devoted to extend the proposed approach with tools providing further automated support during the Reverse Engineering phase
- The network infrastructure will also be extended to let PEP and PDP achieve the available resources of users belonging to a role
- The approach will be also assessed in a larger case study

Thank you for your attention!

Contact Information:
Giuseppe Scanniello
Dipartimento di Matematica e Informatica
Università di Salerno - Via Ponte Don Melillo - 84084 Fisciano (SA)
email: gscanniello@unisa.it