Object-Oriented Analysis and Design Methodology for Secure Web Applications -focused on Role Based Access Control-

Kyung-Soo Joo[1], Jung-Woong Woo[2]

[1] Department of Computer Software Engineering, Soonchunhyang University, Asan 336-745, Republic of Korea
E-mail : gsoojoo@sch.ac.kr
[2] Department of Computer Software Engineering, Soonchunhyang University, Asan 336-745, Republic of Korea,
E-mail : jyone0715@gmail.com

Abstract. In order to develop such web-based application systems efficiently, object-oriented analysis and design methodology is used, and Java EE(Java Platform, Enterprise Edition) technologies are used for its implementation. In addition, security issues have become increasingly important. Consequently, since the security method by Java EE mechanism is implemented at the last step only, it is difficult to apply constant security during the whole process of system development from the requirement analysis to implementation.
In this paper we propose an object-oriented analysis and design methodology emphasized in the security for secure web application systems from the requirement analysis to implementation. Also, the object-oriented analysis and design methodology for the secure web application is applied on an online banking system in order to prove its effectiveness.

Keywords: Object-Oriented Analysis and Design, Web Application, Security, RBAC, Java EE

1 Introduction

Many web-based application systems with various and complicated functions are being requested. In order to develop such web-based application systems efficiently, object-oriented analysis and design methodology is used, and Java EE technologies are used for its implementation [1,2,3]. For this purpose, Java EE-based servlet supports security measures such as role-based access control. But these technologies have no consistency because they are not the ones used as a result of the analysis and design. On that account, the system is very likely to be developed as a web application system which is vulnerable in aspect of security [4,5,6,7,8].

The object-oriented analysis and design methodology with emphasized security is proposed. The methodology provides consistency of security throughout the system development life cycle from requirements analysis till implementation step. In addition, the implementation of security is materialized by using role-based access control which is supported by Java EE-based servlet technology.

2 Related works

CBD(Component Based Development) methodology aims to quickly and flexibly respond to the changes in user's requirements by developing component-based software system[9]. The conceptual model derived by the existing object-oriented analysis and design methodology can generate object-oriented programming code through class diagram. But the consistent analysis and design methodology for security is not being presented [7].

As a security related analysis and design methodology, UML based-development methodology which integrates existing object-oriented analysis and design methodology and security requirement is presented. But the correlation with Java EE is not being presented.

Web application systems are exposed to various risks. In order to avoid these risks, security can be configured in Java EE. The servlet security comprise authentication, authorization, confidentiality, and data integrity [3,10].

3 Object-oriented analysis and design methodology for secure web application

The object-oriented analysis and design methodology for secure web application proposed in this paper has an additional definition of security which has been one of the non-functional requirements in requirement analysis phase, as shown in Fig 1. The added requirement is defined by using UMLsec. In addition, during the phases of system analysis and design, security emphasized analysis and design are presented by using UMLsec. In the final implementation phase, based on the results of the analysis and design, the security requirements are implemented by using Java EE's role-based access control. On the other hand, the functional requirements analysis and system analysis and design are performed by applying the existing CBD methodology.

3.1 Requirement analysis

Defining requirement means the activities to derive and validate the functional and non-functional requirements that users expect from the software[1,11]. Table 1 shows the security requirements. Number 1 is for Administrator's right. Number 2 is for certification. Number 3 is for security requirements from an authorized user. Number 4 is for security requirements of confidentiality and data integrity.

Requirement analysis	Functional	CBD Methodology
	Non-functional	UMLsec Methodology
System analysis and design	Functional	CBD Methodology
	Non-functional	UMLsec Methodology
Implementation	Functional	CBD Methodology or Object-oriented programming
	Non-functional	Java EE's Role-based Access control

Fig 1. Process of Object-oriented analysis and design methodology for secure web application proposed

Table 1. Defining security requirements

Type	Description
Security	1. Administrator has overall authority to access the system through management function, and also create new account, close account, modify balance, cancel transaction, and determine user s rating. 2. Login is required to use the system. 3. Administrator can authorize a particular user to use the system. 4. Functions for data management and protection are required.

Based on the list of user requirements defined in Table 1, a use case is created[1,12]. In case of the use case having security requirements, use case must be extended following the methodology of UMLsec[4]. Table 2 shows an extended use case based on UMLsec methodology for security.

Table 2. Use case having security requirement; Use case for rating set-up

Use Case : Rating Set-up

※ Risks associated with the actor

- User is allowed to check his own information. Administrator can check and modify all user's information.

※ Security required input/output data and security not-required input/output data

Security required I/O	Security not-required I/O
ID	Result output
Password	-

※ Activity of modified system

- User must get a membership.
- User must go through the authentication procedure. Otherwise the User cannot use the system.
- If inputted information is wrong during the authentication process, the system must output an error message.
- Administrator sets User s access rating.
- System provides the output of the result.

Table 3 describes the use case to rating set-up where security is required. And through a use case description sheet, variety of situations, i.e. the use case scenario should be created[11].

Table 3. Use case description for rating set-up

Item	Description		
Name	Rating set-up		
Overview	Administrator can authorize each User's access to the system.		
Relevant Actor	Main Actor	Administrator	
Priority	1	Importance	1(High)
		Difficulty	1(High)
Leading Condition	- Login must be done as Administrator. - User to be configured must have a membership.		
Tailing Condition	- Login status must be maintained. - System shows User s information to Administrator. - System records User's rating.		
Scenario	Basic scenario	Basic scenario between the actor and System	
Non-functional Requirement	Security requirement - Administrator has overall authority to access the system. - Administrator can authorize a particular User to use the system.		

When creating a use case model, individual functions to be provided by the system is represented as use case and the presence outside the system to interact with use case is represented as an actor[11].

3.2 System analysis and design

The target in system analysis and design phases is to identify the components of the system so as to meet the user's requirements. And it should be carried out on the base of requirement model[11]. The process of system analysis and design for the proposed object-oriented analysis and design methodology for secure web applications is shown in Fig 2.

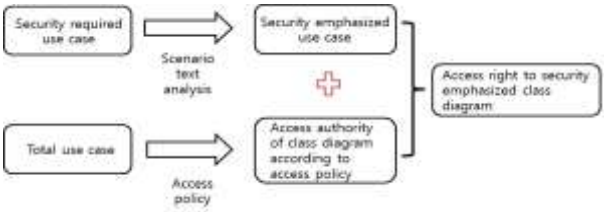

Fig 2. Creating process of security emphasized class diagram depending on access policy

For the use case text analysis, the text of basic scenario contents of use case is analyzed and the classes required for the system operation is extracted[1,11]. The classes are boundary classes, control classes, and entity classes[11].

Next, the individual actor's access right for each use case should be created[3,4]. The created access policy indicates the access right for class diagram. Table 4 defines the access policy for a part of on-line banking system use case.

Table 4. Use case access policy according to an actor

	Customer	Staff	Administrator
Membership	X	X	X
Login	X	X	X
...	...	...	...
Rating set-up	-	-	X
Legend : Full right(X), Partial right(P), No right(-)			

After the access policy creation activities, the activity for creation of analysis class diagram should be carried out by analyzing the text of use case scenario[11]. In other words, the activity is to derive the classes and define the relationship between classes and derived classes.

The classes derived from the use case with security requirement is security emphasized classes and each class is created for the access right depending on access policy following UMLsec methodology and using <<secrecy>> stereo type, refer to Table 3

For detailed analysis class diagram, based on the security emphasized class diagram which is derived from the previous activity, the text of use case scenario should be additionally analyzed and the characteristic and the operation of each analysis class should be defined[1, 11].

MVC pattern is applied to the detailed analysis class diagram as follows. The class using <<entity>> stereo type corresponds to the Model. The class using <<boundary>> stereo type represents the View and it can be implemented by JSP and so on. The class using <<control>> stereo type represents Controller and it can be implemented by Servlet and so on.

The class using <<secrecy>> stereo type is security emphasized class. If it is used with <<control>> and <<boundary>>, Java EE's role-based access control can implement it.

3.3 Implementation

In User control class which is related to rating set-up use case, <<control>> and <<secrecy>> are used. Thus role should be defined to apply security mechanism of Java EE. Table 5 shows how to define role for authentication and authorization.

Table 6 shows how to implement authentication. There are four methods available: BASIC, DIGEST, CLIENT-CERT, and FORM. In this paper, authentication is implemented by FORM. In case of <form-login-page> and <form-error-page>, if authentication is FORM-based, it can be defined to show random page created by the developer.

- Tomcat-user.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
<role rolename="admin"/>
<role rolename="customer"/>
<user username="admin"
  password="admin1234"
  roles="admin"/>
<user username="customer"
  password="customer1234"
  roles="customer"/>
</tomcat-users>
```

- web.xml

```
<login-config>
<auth-method>FORM</auth-method>
<form-login-config>
<form-login-page>/login.jsp</form-login-page>
<form-error-page>/loginerror.html</form-error-page>
</form-login-config>
</login-config>
```

Table 5. Role defining

Table 6. Implemented authentication

Table 7 and Table 8 show authorization steps. For the request to Servlet, an appropriate role should be mapped to deployment descriptor. And accessible resource and usable HTTP method should be specified. As the customer management page, such as Rating set-up, can only be accessed by Administrator, the access rights for that page is configured as follows.

Table 7. Role registration

- web.xml

```
<security-role>
<role-name>admin</role-name>
<role-name>customer</role-name>
</security-role>
```

Table 8. Defining resource and method restriction

- web.xml

```
<security-constraint>
<web-resource-collection>
<web-resource-name>test web resource
</web-resource-name>
<url-pattern>/admin/Member.jsp</url-pattern>
<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>
<auth-constraint>
<role-name>admin</role-name>
</auth-constraint>
</security-constraint>
```

4. Verification of the Object-Oriented Analysis and Design Methodology for Secure Web Applications

A type of attack attempted by an unauthorized user, who failed to pass the authentication but fakes as an authorized user, can be defended as shown in Fig 3. And another type of attack attempted by a user who fakes his/her rating can be defended through the authentication process as shown in Fig 4.

Fig 3. Login Error page

Fig 4. Access rating Error page

Also, the effectiveness of the object-oriented analysis and design methodology for secure web application system was confirmed by defending against various attacks such as to modify user's important information or to sneak data, through the features of confidentiality and data integrity as shown in Fig 5.

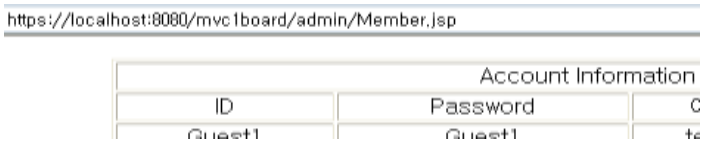

Fig 5. Customer management page

5. Conclusion

This study suggests an object-oriented analysis and design methodology for secure web application system.

The object-oriented analysis and design methodology for secure web application system offers a consistent analysis and design method that was not supported by existing object-oriented analysis and design methodologies.

References

1. Brett D. McLaughlin, Gary Pollice, David West.: Head First Object Oriented Analysis & Design. Hanbit Media, Seoul (2007)
2. Han Jeong-Su, Kim Gwi-Jeong, Song Yeong-Jae.: Introduction to UML : Object-Oriented Design as in a friendly learning. Hanbit Media, Seoul (2009)
3. Joo Kyung-Soo, Woo Jung-Woong.: A Development of the Unified Object-Oriented Analysis and Design Methodology for Security-Critical Web Applications Based on Object-Relational Database – Forcusing on Oracle 11g -, Journal of The Korea Society of Computer and Information, 17, pp. 169-177 (2012)
4. G.Popp, J. Jurjens, G.Wimmel, R. Breu.: Security-Critical System Development with Extended Use Case, Asia-Pacific Software Engineering Conference (2003)

5. Madan, s.: Security Standards Perspective to Fortify Web Database Applications From Code Injection Attacks. Information Conference on Intelligent System, Modelling and Simulation(ISMS), 10, pp. 226-230 (2010)
6. lqra Basharat, Farooque Anam, Abdul Wahab Muzaffar.: Database Security and Encryption; A Survey Study, International Journal of Computer Application, 47, pp. 28-34(2012)
7. Cho Wan-Su.: UML2&UP Object-Oriented Analysis&design. Hongrung Publishing, Seoul (2005)
8. David Basin, Jürgen Doser and Torsten Lodderstedt.: Model Driven Security; from UML Models to Access Control Infrastructures, ACM Transaction on Software Engineering and Methodology(TOSEM), 15, pp. 39-91 (2006)
9. Jun Byung-Sun.: CBD, WHAT&HOW. Wowbooks, Seoul (2005)
10. Kathy Sierra, Bert Bates, Bryan Basham.: Head First Servlet & JSP. Hanbit Media, Seoul (2009)
11. Chae Heung-Seok.: Object-oriented CBD Project for UML and Java as learning. Hanbit Media, Seoul (2009)