International Journal of Security and Its Applications
Vol.8, No.1 (2014), pp.71-80
http://dx.doi.org/10.14257/ijsia.2014.8.1.07

Development of Object-Oriented Analysis and Design Methodology for Secure Web Applications

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

[1]*Department of Computer Software Engineering, Soonchunghyang University, 336-745 Asan, Requblic of Korea*
[2]*Department of Computer Software Engineering, Soonchunghyang University, 336-745 Asan, Requblic of Korea*
[1]*gsoojoo@sch.ac.kr*, [2]*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.

Therefore, this paper suggests an object-oriented analysis and design methodology emphasized in the security for secure web application systems from the requirement analysis to implementation. The object-oriented analysis and design methodology adopts UMLsec, the modeling language with an emphasis on security for the requirement analysis and system analysis and design with regard to security. And for its implementation, RBAC(Role Based Access Control) of servlet from Java EE technologies is used. 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 [1-3]. 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 [4-8].

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

ISSN: 1738-9976 IJSIA

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 Figure 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.

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

Figure 1. Process of Object-oriented Analysis and Design Methodology for Secure Web Application Proposed

3.1. Requirement Analysis

3.1.1. Requirement List-up

Defining requirement means the activities to derive and validate the functional and non-functional requirements that users expect from the software [1,11,12]. Table 1 is the requirements list of a part of the on-line banking system that includes the security requirement definition of non-functional requirements.

Table 1. Requirement List for On-line Banking System

1. User can use an inquiry service.
2. The inquiry service provides balance checking, transaction list checking, record checking and download function.
3. User can use fee payment service and various taxes payment function.
4. User can use transaction service.
5. Transaction service includes functions such as money transfer and so on
6. 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.
7. Permission to use the system can be given for a particular user.
8. Login is required to use the system.
9. Functions for data management and protection are required.

Table 2 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.

Table 2. 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.

3.1.2. Creating Use Case

Based on the list of user requirements defined in Table 1, a use case is created [1, 11]. In case of the use case having security requirements, use case must be extended following the methodology of UMLsec [5]. Table 4 shows an extended use case based on UMLsec methodology for security.

Table 3. Use Case List

Use case	Description
Membership	User can get a membership to use the system.
Login	User can log in to use the system.
…	…
Rating set-up	Administrator can set User's access rating.

Table 4. 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.

3.1.3. Detailed Use Case Model

Table 5 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, 12]. Table 6 describes the basic scenario of use case to rating set-up.

Table 5. 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.		

Table 6. Basic Scenario of Rating Set-up Use Case

1. User must have a membership. 2. User inputs ID and password on login screen and clicks login button. 3. System shows Administrator a screen. Administrator selects rating on Administrator's screen. 4. User's rating can be checked on the rating set-up screen. In order to modify the rating, click the rating button. 5. System shows detailed rating information screen. ※ Detailed rating information screen : ID, name and rating 6. Administrator modifies the rating and clicks confirmation button. The system records the modified data and updates the detailed rating information screen. 7. To return to the previous screen, click the cancel button.

3.1.4. Use Case Model Creation

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, 12]. Figure 2 shows creating a use case model for on-line banking system.

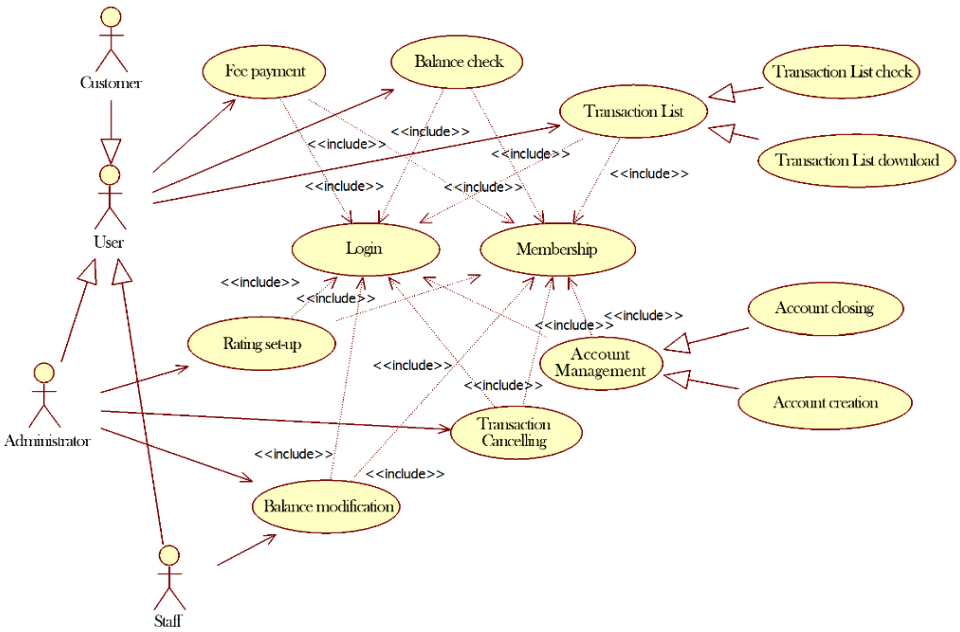

Figure 2. Use Case Model for On-line Banking System

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 Figure 3.

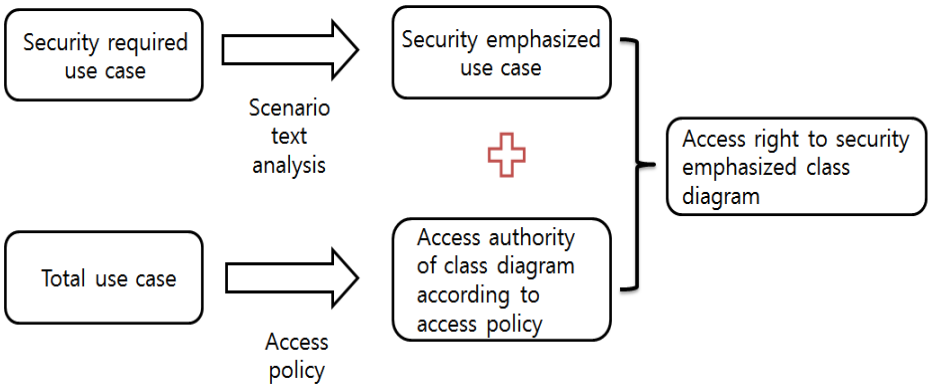

Figure 3. Creating Process of Security Emphasized Class Diagram Depending on Access Policy

3.2.1. Analysis of Use Case Text

For the use case text analysis, the text of basic scenario contents of use case which is written on the base of the information obtained from the user requirements is analyzed and the classes required for the system operation is extracted [1, 11].

The classes that can be extracted through the analysis of the text are a boundary class, a control class, and an entity class [11].

3.2.2. Analysis of Use Case Text

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 7 defines the access policy for a part of on-line banking system use case.

Table 7. Basic Scenario of Rating Set-up Use Case

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

3.2.3. Creating Analysis Class Diagram

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 7.

3.2.4. Detailed Analysis Class Diagram

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 [11, 13]. Figure 4 shows a detailed analysis class diagram using <<secrecy>> stereo type.

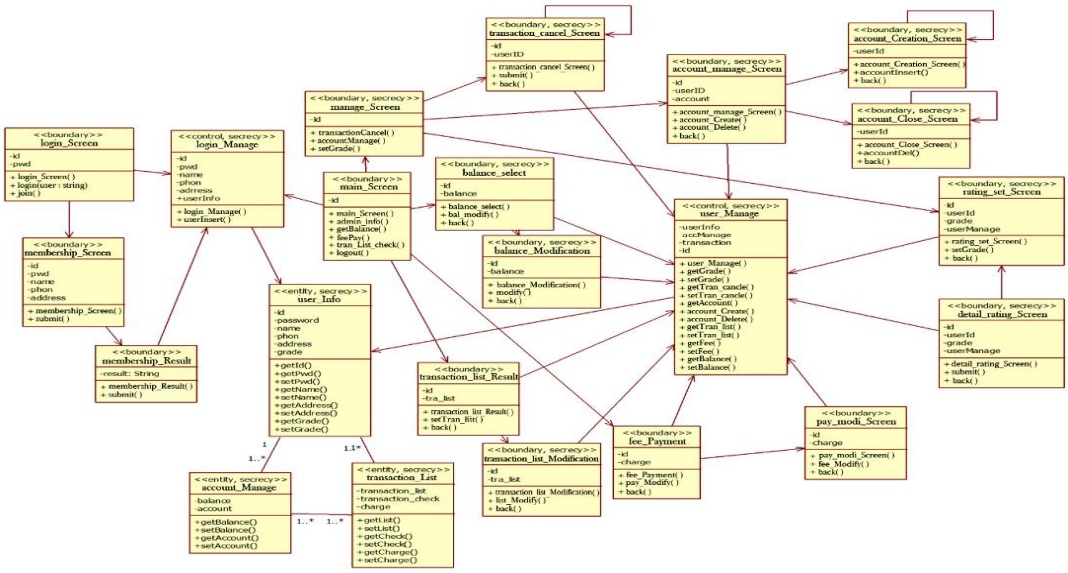

Figure 4. Detailed analysis class diagram for on-line banking system

3.2.5. Java EE-based MVC pattern application

MVC pattern is applied to the detailed analysis class diagram as follows.

① The class using <<entity>> stereo type corresponds to the Model.

② The class using <<boundary>> stereo type represents the View and it can be implemented by JSP and so on.

③ The class using <<control>> stereo type represents Controller and it can be implemented by Servlet and so on.

④ 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

3.3.1. Java EE-based Role Based Access Control

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 8 shows how to define role for authentication and authorization.

Table 9 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.

Table 8. Role Defining

```
- 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>
```

Table 9. Implemented Authentication

```
- 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 10 and Table 11 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 Figure 5. And another type of attack attempted by a user who fakes his/her rating can be defended through the authentication process as shown in Figure 6.

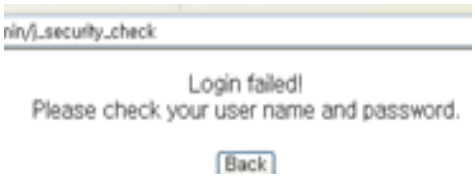

Figure 5. Login Error Page

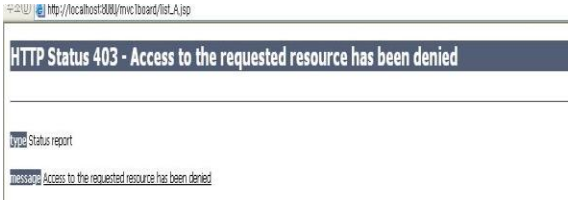

Figure 6. Access Rating Error Page

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 Figure 7.

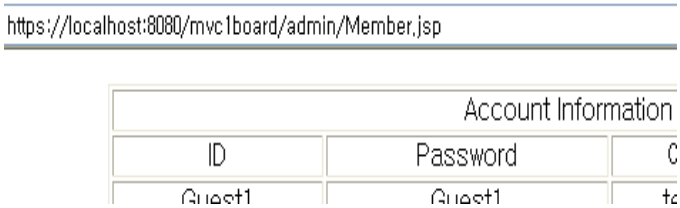

Figure 7. Customer Management Page

5. Conclusion

This study suggests an object-oriented analysis and design methodology for secure web application system. For such purpose, a security emphasized modeling language, UMLsec was used and Java EE's role based access control was used for the implementation.

Therefore, 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. In addition, the correlation with Java EE that was not provided by UMLsec is provided through role-based access control. Thus, the correlations among existing object-oriented analysis and design methodologies, security, and Java EE are presented to enable object-oriented analysis and design for the whole process of system development.

It concludes that the effectiveness of the object-oriented analysis and design methodology for secure web application system was proved by successfully applying it to the on-line banking system development.

References

[1] B. D. McLaughlin, G. Pollice and D. West, "Head First Object Oriented Analysis & Design", Hanbit Media, Seoul, **(2007)**.

[2] J.-S. Han, G.-J. Kim and Y.-J. Song, "Introduction to UML: Object-Oriented Design as in a friendly learning", Hanbit Media, Seoul, **(2009)**.

[3] K.-S. Joo and J.-W. Woo, "A Development of the Unified Object-Oriented Analysis and Design Methodology for Security-Critical Web Applications Based on Object-Relational Database – focusing on Oracle 11g", Journal of The Korea Society of Computer and Information, vol. 17, **(2012)**, pp. 169-177.

[4] G. Popp, J. Jurjens, G. Wimmel and R. Breu, "Security-Critical System Development with Extended Use Case", Asia-Pacific Software Engineering Conference, **(2003)**.
[5] S. Madan, "Security Standards Perspective to Fortify Web Database Applications from Code Injection Attacks", Information Conference on Intelligent System, Modelling and Simulation(ISMS), vol. 10, **(2010)**, pp. 226-230.
[6] L. Basharat, F. Anam and A. Wahab Muzaffar, "Database Security and Encryption; A Survey Study", International Journal of Computer Application, vol. 47, **(2012)**, pp. 28-34.
[7] W.-S. Cho, "UML2&UP Object-Oriented Analysis & design", Hongrung Publishing, Seoul, **(2005)**.
[8] D. Basin, J. Doser and T. Lodderstedt, "Model Driven Security; from UML Models to Access Control Infrastructures", ACM Transaction on Software Engineering and Methodology (TOSEM), vol. 15, **(2006)**, pp. 39-91.
[9] B.-S. Jun, "CBD, WHAT & HOW", Wowbooks, Seoul, **(2005)**.
[10] K. Sierra, B. Bates and B. Basham, Head First Servlet & JSP. Hanbit Media, Seoul, **(2009)**.
[11] H.-S. Chae, "Object-oriented CBD Project for UML and Java as learning. Hanbit Media", Seoul, **(2009)**.
[12] M. Su, F. Li, G. Shi and L. Li, "An Action Based Access Control Model for Multi-level Security", IJSIA, vol. 6, **(2012)**, pp. 359-366.
[13] A. Maamir, A. Fellah and L. A. Salem, "Fine Granularity Access Rights for Information Flow Control in Object Oriented Systems", IJSIA, vol. 2, **(2008)**, pp. 81-92.

Authors

Kyung-Soo Joo, he received the B.S. degrees in Depart of Mathematics from Korea University, Korea, in 1980 and M.S. Ph.D. degrees in Department of Computer Science from Korea University, Korea, in 1986 and 1993. Now he is a professor in Department of Computer Software Engineering from Soonchunhyang University, Korea. His research interests include Database System, Object-Oriented System, BigData databases, etc.

Jung-Woong Woo, he received the B.S. degrees in Depart of Computer Software Engineering from Soonchunghyang University, Korea, in 2013. He is now a M.S. candidate in Department of Computer Software Engineering from Soonchunhyang University, Korea. His research interests include Database System, UML, Object-Oriented System, Cloud Databases, Big Databases, etc.

