한국컴퓨터정보학회 동계학술대회 논문집 제21권 제1호 (2013. 1)

웹 응용시스템 개발을 위한 보안을 고려한 통합 분석 · 설계 방법론 개발

우정웅°, 김동섭*, 주경수**
°순천향대학교 컴퓨터소프트웨어공학과
*카네기멜론대학교 의료공학과
**순천향대학교 컴퓨터소프트웨어공학과
e-mail : jyone0715@gmail.com°, peter1114@gmail.com*, gsoojoo@sch.ac.kr*

A Development of the Unified Object-Oriented Analysis and Design Methodology for Security-Critical Web Applications Based on Relational Database

Jung-Woong Woo°, Dong-Seob Kim*, Kyung-Soo Joo**
°Dept. of Computer Software Engineering, Soonchungyang University
*Dept. of Biomedical Engineering, Carnegie Mellon University
**Dept. of Computer Software Engineering, Soonchungyang University

● 요 약 ●

응용시스템 개발 과정에 있어서 중요하고 핵심을 이루는 작업은 분석과 설계 작업이며 아울러 대부분의 응용시스템은 데이터베이스 기반으로 구축된다. 또한, 응용시스템들은 외부 공격에 쉽게 노출되기 때문에 보안과 관련된 처리 과정 역시 중요하다. 하지만 이러한 보안은 대부분 개발 마지막 과정에서 고려하기 때문에 보안에 취약한 응용시스템들이 개발될 가능성이 매우 높다. 따라서 개발 초기에 보안을 반영한 분석 및 설계 과정이 매우 중요하다.

Java EE는 웹 응용시스템을 위한 보안 방안을 제공하고 아울러 관계형 데이터베이스도 보안을 위하여 역할기반 접근제어를 지원하고 있지만 관계형 데이터베이스 및 Java EE의 역할기반 접근제어를 활용하는, 요구사항 수집부터 구현까지 개발 단계 전체에 걸친 일관된 개발방법론은 전무한 실정이다. 따라서 본 논문에서는 보안 요구사항을 요구사항 수집부터 분석 및 설계 그리고 마지막 구현 단계까지 반영하여 Java EE 기반의 웹 응용시스템을 개발하기 위한, 보안을 고려한 일관된 통합 분석·설계 방법론을 제안한다.

키워드: 객체지향 분석 · 설계(Object-Oriented Analysis Design), RBAC(Role Based Access Control), Java EE(Java Platform, Enterprise Edition), 보안(Security), 관계형 데이터베이스 설계(Relational Database Design)

I. 서 론

IT기술의 급속한 발전으로 인해 Java EE(Java Platform, Enterprise Edition) 기반의 웹 응용소프트웨어 시스템이 많이 개발되고 있다[1,2]. 그러나 웹 응용시스템 개발을 위한 객체지향 분석·설계방법론과 관계형 데이터베이스 설계를 위한 방법론들이 따로 존재하여, 일관된 웹 응용시스템을 개발하기 어렵다.

이에 따라 본 논문에서는 기존의 객체지향 분석·설계방법론과 관계형 데이터베이스 설계방법을 기반으로, 보안 요구사항을 요구사항 수집부터 분석·설계 그리고 구현 단계까지, 전 개발단계에 걸쳐 보안에 대한 일관성을 제공하는 통합 객체지향 분석·설계 방법론을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안한 방법론의 이해를 돕기 위한 관련 연구들을 소개하고, 3장에서는 제안한 방법론의 적용, 4장에서는 결론을 제시한다.

II. 관련 연구

1. 객체지향 분석 · 설계 방법론

객체지향 분석·설계 방법론 중 대표적인 RUP(Ra-tional Unified Process)의 특징은 유스케이스 기반, 아키텍처 중심, 반복 및 점증적

이며, 도메인 모델, 유스케이스 모델, 분석 모델, 설계 모델 그리고 구현 모델 등으로 작성된다는 것이다[3]. 다만 RUP는 보안에 대한 일관된 분석·설계 방법론은 제시하지 못하고 있다.

2. 관계형 데이터베이스 설계 방법론

대표적인 관계형 데이터베이스 설계방법론으로는 정보공학방법론(Information Engineering Methodology)이 있으며, 데이터 중심의 업무절차 및 환경변화에 유연한 장점을 가지고 있는 안정정인 방법론 중의 하나이다[10]. 그러나 객체지향 분석·설계 방법론과 상호 연관성을 제공하지 못하고 있으며, 보안과 관련된 일관성을 제공하지 못하고 있다.

3. UMLsec을 이용한 보안 유스케이스 모델링

보안과 관련한 분석·설계 방법으로는, 기존의 객체지향 분석·설계 방법론과 보안 요구사항을 통합한 UML 기반의 개발방법론이 제시되었다[4]. 이 연구에서는 확장된 UMLsec을 이용해서 보안이 중요한 응용시스템 개발을 위한 일관된 객체지향 분석·설계 방법론을 제시하고는 있지만, 관계형 데이터베이스 및 Java EE와의 상호 연관성은 제공하지 못하고 있다.

4. J2EE 기반의 웹 보안

웹 응용 시스템들은 다양한 위험에 노출되어 있다. 이러한 위험을 막기 위해 서블릿에서 보안을 설정할 수 있으며, 서블릿 보안의 4요소는 인증, 인가, 비밀보장, 데이터 무결성으로 이뤄진다. 이에 Java EE에서의 인증은 BASIC, DIGEST, CLIENT-CERT, FORM과 같이 4가지 인증 방법이 존재한다[11].

한편, 웹 기반의 응용시스템은 MVC 패턴을 주로 사용하여 개발하고 있다.

III. Java EE 기반의 보안을 고려한 통합 분석 · 설계 방법론

본 논문에서 제안한 보안을 고려한 통합 객체지향 분석·설계 방법론은, 그림1과 같이 요구사항 정의 단계에서 보안에 대한 요구사항을 추가하였으며, 분석 및 설계 단계에서는 UMLsec을 이용한 보안 요구사항을 반영하였고 아울러 관계형 데이터베이스 설계 방법론과 통합하였다. 또한 마지막 구현 단계에서는 보안 요구사항의 설계 결과를 바탕으로 Java EE의 역할기반 접근제어와 관계형 데이터베이스의 역할기반 접근제어를 이용하여 보안에 대한 요구사항을 구현하였다.

1. 통합 객체지향 분석 · 설계 방법론

1. 요구사항 정의

1.1 요구사항 리스트 작성

요구사항 정의 단계는 사용자로부터 요구사항을 도출하고 이를 분석하고 요구사항 리스트를 작성하는 활동으로 구성된다[1]. 표 1은 가로세로 퍼즐 시스템의 요구사항 리스트에 해당한다.

표 1. 가로세로 퍼즐 시스템을 위한 요구사항 리스트
Table 1. Requirement list of 'Horizontal and Vertical Puzzle System'

1.학생은 원하는 챕터의 퍼즐을 선택 할 수 있어야 한다.
2.학생은 챕터 선택으로 되돌아 갈수 있어야 한다.
3.학생은 원하는 챕터의 퍼즐이 보여야 한다.
4.학생은 퍼즐의 한 칸을 선택하면 문제를 볼 수 있어야 한다.
5.학생은 문제에 답을 쓸 수 있어야 한다.
6.학생에게 힌트를 제공해야 한다.
7.학생은 문제를 풀지 않을 수 도 있다.
8.퍼즐은 학생이 답을 맞추면 답을 보여줘야 한다.
9.교수만 새로운 챕터의 퍼즐을 만들 수 있어야 한다.

1.2 요구사항 리스트 작성

유스케이스는 시스템이 어떤 일을 수행하기 위해 거쳐야 하는 단계들을 말하며, 또한 새로 만들 시스템이나 소프트웨어 변경사항에 대한 요구사항을 찾아내는 방법이다[1].

표1에서 작성된 사용자 요구사항 리스트를 기반으로, 보안이 고려된 유스케이스를 작성한다. 또한 보안이 요구되는 유스케이스의 경우에는 유스케이스를 확장해야 된다. 다음 표2는 보안이 요구되는 '사용신청'에 대한 확장된 유스케이스이다.

표 2. '사용신청' 을 위한 유스케이스
Table 2. Usecase of 'Request'

Use Case : 사용신청

Actor와 관련된 위험성
- 별도의 계정구분이 없기 때문에 학생 계정을 가지고 있는 사용자가 교수로 위장하여 접근할 수 있다.

Security-Critical과 uncritical 입·출력 데이터

Security-Critical I/O	uncritical I/O
아이디	-
패스워드	-

변경된 시스템의 행동
- 교수와 학생은 사용신청 시 신분이 구분될 수 있도록 별도의 계정으로 가입된다. 즉, 사용허가 유스케이스를 포함하여 시스템으로부터 사용자 구분에 맞는 인증을 받게 되며, 사용신청 결과 화면을 보여준다.

1.3 유스케이스 모델 상세화

유스케이스 상세화 활동에서는 직전 활동에서 도출된 각 유스케이스별로 개요, 관련 액터, 우선순위, 선행/수행 조건, 시나리오 비기능적 요구사항을 정의하며, 도출된 명세서를 활용하여 시나리오를 작성하는 과정이다[7]. 다음 표 3은 보안이 요구된 '사용신청' 유스케이스 명세서에 해당하며, '사용신청' 유스케이스의 기본 시나리오는 표 4와 같다.

표 3. '사용신청' 을 위한 유스케이스 명세서
Table 3. Usecase Specification of 'Request'

항 목	설 명		
이름	사용신청		
개요	학생과 교수는 계정에 대한 사용신청을 한다.		
관련 액터	주액터	학생, 교수	
우선 순위	1	중요도	1(상)
		난이도	3(하)
선행 조건	학생과 교수는 아이디와 패스워드를 입력하고 확인 버튼을 누른 상태이어야 한다.		
후행 조건	사용신청 확인 결과를 보여준다.		
시나리오	기본 시나리오	학생 및 교수는 아이디를 입력한다.	
	대안 시나리오	조건에 맞지 않는 아이디 및 패스워드를 입력했을 경우에 대한 경고문을 확인한다.	
비기능적 요구사항	교수만이 퍼즐을 생성할 수 있다.		

표 4. '사용신청' 을 위한 기본 시나리오
Table 4. Basic Scenario of 'Request'

1.사용자는 사용신청 버튼을 누른다.
2.아이디와 패스워드를 입력한다.
  2.1 조건에 맞지 않는 데이터를 입력하지 않을 경우 경고문 확인 후 다시 입력한다.
  2.2 되돌아가기 버튼을 누른다.
3.확인 버튼을 누른다.
4.사용신청 결과를 화면에 보여준다.

1.4 유스케이스 모델 작성

유스케이스 모델 작성은 시스템이 제공할 개별 기능을 유스케이스로 표현하고, 유스케이스와 상호작용을 하는 시스템 외부의 존재를 액터로 표현한다. 그리고 유스케이스 모델의 시각적인 표현을 위해 UML의 유스케이스 다이어그램을 사용하며, 액터와 유스케이스 간의 연관 관계를 표현함으로써 어떤 액터가 어떤 유스케이스를 이용하는지를 기술한다[7]. 다음 그림 2는 가로세로 퍼즐 시스템의 유스케이스 모델 작성을 보여준다.

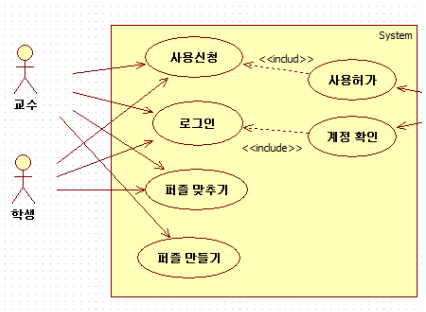

그림 2. '가로세로 퍼즐 시스템' 을 위한 유스케이스 모델
Fig. 2. Usecase Model of 'Horizontal and Vertical Puzzle System'

2. 보안을 고려한 분석 · 설계 단계

2.1 유스케이스 본문 분석

유스케이스 본문 분석은 사용자 또는 고객으로부터 얻은 요구사항 정보들을 토대로 그 내용을 분석하여 소프트웨어 시스템에 필요한 클래스들을 추출해 내는 작업을 말한다[1]. 다음 그림 3은 보안이 고려된 '사용신청' 유스케이스의 본문 분석에 해당한다.

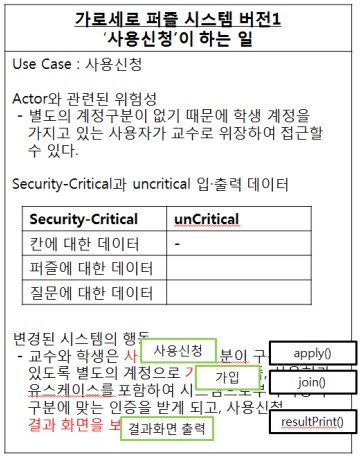

그림 3. '사용신청' 을 위한 본문 분석
Fig. 3. Text Analysis of 'Request'

2.2 분석 클래스 모델 작성

접근정책 작성 활동 이후, 분석 클래스 모델의 작성 활동은 유스케이스의 명세서를 분석해 클래스 다이어그램을 작성하는 활동이다[7]. 또한 확장된 유스케이스 의 본문분석 뿐만 아니라, 별도의 접근정책을 활용하여 보안이 요구되는 클래스들을 구별할 수 있다. 보안이 요구되는 클래스는 <<secrecy>> 태그를 사용한다. 다음 그림 4는 '가로세로 퍼즐 시스템'의 상세화된 클래스 다이어그램이다.

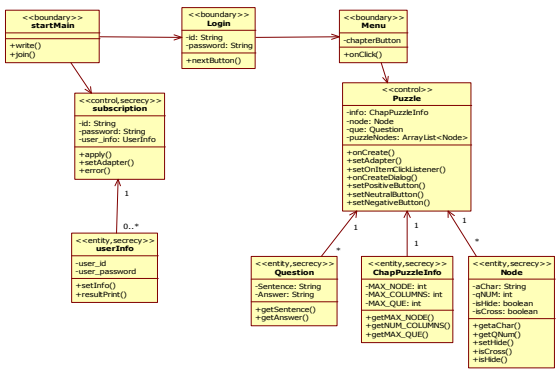

그림 4. '가로세로 퍼즐 시스템' 의 상세화된 분석 클래스 모델
Fig. 4. Detailed Analysis Class Model of 'Horizontal and Vertical Puzzle System'

3. 구현

3.1 Java EE의 역할기반 접근제어

웹 기반의 응용소프트웨어 개발을 위해 상세화된 분석 클래스 다이어그램에 MVC 패턴을 적용한다. 따라서 그림6의 상세화된 클래스 다이어그램은 각 스테레오 타입에 따라 다음과 같은 조건을 따른다.

① <<entity>> 타입을 사용한 클래스는 Model로서 데이터베이스의 스키마로 변환시킨다.
② <<boundary>> 타입을 사용한 클래스는 View로서 JSP 등으로 구현한다.
③ <<control>> 타입을 사용한 클래스는 Con-troller로서 서블릿 등으로 구현한다.
④ <<secrecy>> 타입을 사용한 클래스는 보안이 고려되어야만 하는 클래스이며, <<entity>> 타입과 같이 사용되었다면 데이터베이스의 역할기반 접근제어를 이용하여 보안을 적용한다. 또한 <<control>> 타입과 같이 사용되었다면 Java EE의 보안 메커니즘을 적용한다.

본 논문에서 사용된 예에서는 subscription 클래스 다이어그램이 <<control>>과 <<secrecy>>가 적용되어 있기 때문에 Java EE 기반의 보안 메커니즘을 적용하기 위해 role을 정의한다. 다음 표 5와 같이 인증과 인가를 통해 Java EE 기반의 보안 메커니즘을 적용할 수 있다.

표 5. 인증 구현
Table 5. Certification Materialization

```
- Tomcat-user.xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="professor"/>
    <role rolename="manager"/>
    <user username="professor"
      password="1234"
      roles="professor manager"/>
 </tomcat-users>
- web.xml
<security-role>
     <role-name>professor</role-name>
</security-role>
<security-constraint>
  <web-resource-collection>
       <web-resource-name>test web resource
   </web-resource-name>
       <url-pattern>/*</url-pattern>
       <http-method>GET</http-method>
       <http-method>POST</http-method>
  </web-resource-collection>
    <auth-constraint>
       <role-name>professor</role-name>
  </auth-constraint>
<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>
```

4. 관계형 데이터베이스 설계

4.1 관계형 데이터베이스 설계방법

관계형 데이터베이스 설계방법은 O-R(Object-Relational)매핑을 통하여 진행한다. 표 6의 변환방법을 이용하여 엔티티 클래스를 관계형 데이터베이스 스키마로 변환한다[8].

표 6. 관계형 데이터베이스 스키마로의 변환방법(O-R매핑)
Table 6. Transformation of Relational Database Schema (O-R Mapping)

① 클래스는 테이블이 됨.
② 클래스의 속성(attribute)은 테이블의 열(column)이 됨.
③ 클래스의 속성 타입은 테이블의 열 타입이 됨.
④ 일반화가 없는 클래스를 위해서는 integer 기본키를 생성하고, {oid}를 위해서는 기본키 제약 조건에 {oid} 태그 열을 추가함.
⑤ 자식 클래스(Subclasses)들은, 각 부모 클래스의 키를 기본키와 외부키 제약조건에 추가함.
⑥ 속성에 {nullable} 태그가 있으면 테이블 속성에 NULL 또는 NOT NULL을 추가함.
⑦ 속성이 초기 값을 가지면, 열에 DEFAULT 문을 추가함.
⑧ 연관 클래스들은, 각 역할-실행 테이블에 대한 기본키를 기본키와 외부키 제약 조건에 추가함.
⑨ 만일 {alternate oid = 〈n〉} 태그이면, UNIQUE 제약 조건에 대한 열을 추가함.
⑩ 각 명시된 제약에 대해 CHECK를 추가함.
⑪ 0..1, 1..1 규칙의 연관 관계에서 참조하는 테이블에 외부키를 생성함.
⑫ 집합 테이블(CASCASE와 같이)의 외부키를 갖는 복합집합을 위해서 기본키를 생성한다 : 기본키를 위해 추가적인 열(column)을 추가함.
⑬ 이진 연관 클래스를 적당한 "N"쪽 테이블로 이동함으로써 최적화 함.
⑭ 연관 클래스가 아닌 3원 연관은 N : N에 대한 테이블로 생성함.
⑮ N : N, 3원 연관에서 역할-실행 테이블의 키로부터 기본키와 외부키 제약 조건을 생성함.
⑯ 연관 클래스 없는 다대다(many-to-many) 연관을 위해 기본키와 외부키를 생성함.

4.2 변환과정

그림 5는 상세화된 분석 클래스 모델에서 엔티티 클래스들 간의 클래스 다이어그램이다. [그림 7]의 ChapPuzzleInfo 클래스는 표 6의 변환 방법 ①과 ② 그리고 ④에 따라 'pid' 속성을 추가로 갖게 되고, 변환 방법 ⑪에 의해 표 7과 같이 관계형 데이터베이스 스키마로 변환된다.

또한 그림 5에서 보안이 요구되는 엔티티 클래스들은 ChapPuzzleInfo, Node, Question 클래스이며, 접근정책에 따라 각 액터에 대한 접근권한을 역할기반 접근제어를 통해 설정할 수 있다. 표 8은 보안이 적용된 교수에 대한 접근 권한이다.

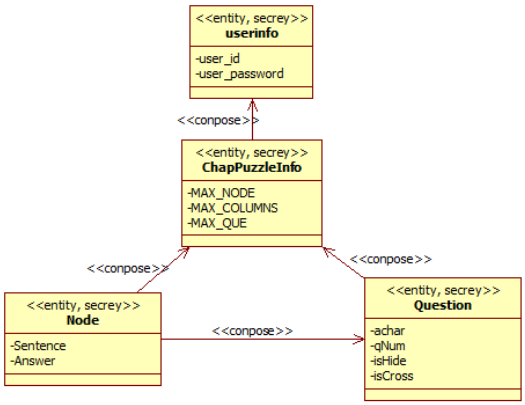

그림 5. '가로세로 퍼즐 시스템' 을 위한 클래스 다이어그램
Fig. 5. Class Diagram of 'Horizontal and Vertical Puzzle System'

표 7. '퍼즐정보' 테이블 스키마
Table 7. 'Puzzle Info' Table Schema

```
CREATE TABLE puzzle_info  (
pid INTEGER PRIMARY KEY,
totalcellnum INTEGER NOT NULL,
colcellnum INTEGER NOT NULL,
quizn INTEGER NOT NULL,
quiznum INTEGER REFERENCE puzzle_quiz,
qnum INTEGER REFERENCE puzzle_node,
CONSTRAINT puzzleinfo_PK PRIMARY KEY(pid, quiznum,
qnum) );
```

표 8. '교수' 에 대한 접근 권한 스키마
Table 8. Access policies Schema of 'Professor'

```
CREATE ROLE pro_entry;
GRANT pro_entry T user_id;
GRANT ALL ON
puzzle_info TO pro_entry;
GRANT ALL ON
puzzle_node TO pro_entry;
GRANT ALL ON
puzzle_quiz TO pro_entry;
GRANT ALL ON
```

IV. 결 론

본 논문에서는 관계형 데이터베이스를 이용한 Java EE 기반의 웹 응용시스템 개발을 위한, 보안을 고려한 통합 객체지향 분석·설계 방법론을 개발하였다. 이를 위하여 보안에 대한 요구사항을 수집하고, 수집된 요구사항을 UMLsec을 이용하여 분석·설계 과정에 반영하였으며 아울러 그 수행결과를 관계형 데이터베이스 설계 방법론과 통합하여 Java EE 및 관계형 데이터베이스의 역할기반 접근제어를 이용하여 구현하였다.

본 논문에서 제시한 통합 객체지향 분석·설계 방법론은 기존의 객체지향 분석·설계 방법론이 제시하지 못했던 보안에 대한 일관된 분석·설계 방법을 제공하고 있으며, 관계형 데이터베이스 설계 방법론의 약점이었던, 객체지향 분석·설계 방법론과의 연관성과 개발과정 전체에 걸친 보안에 대한 일관성을 제공하고 있다.

본 연구에서 제안한 통합 개발방법론은 관계형 데이터베이스를 사용한 Java EE 기반의 보안이 요구되는 웹 응용시스템을 개발하는데 사용하였다.

참고문헌

[1] Brett D. McLaughlin, Gary Pollice, David West, Head First Object Oriented Analysis & Design, Hanbit Media. Inc, pp. 96-103, 2007.

[2] Han Jeong-Su, Kim Gwi-Jeong, Song Yeong-Jae, Introduction to UML : Object-Oriented Design as in a friendly learning, Hanbit Media. Inc, pp. 58-66, 2009.

[3] Cho Wan-su, "UML 2 & UP Object-Oriented Analysis&design", pp.189-205, Hongrung Publishing Company, 2005.

[4] Joo Kyung-Soo, "Problems of Relational Database Design", Soonchunhyang University, Vol. 15, No.1, pp.235-243, 1992.

[5] G.Popp, J. Jurjens, G.Wimmel, R. Breu, "Security-Critical System Development with Extended Use Case", Asia-Pacific Software Engineering Conference, 5-1 self, 2003

[6] Kathy Sierra, Bert Bates, Bryan Basham, Head First Servlet & JSP, Hanbit Media. Inc, pp. 683-721, 2009.

[7] Chae Heung-Seok, Object-oriented CDB Project for UML and Java as learning, Hanbit Media. Inc, pp. 290-960, 2009.

[8] Joo Kyung-Soo, Joo Do-Hyung, "Development of Integrated Design Methodology for Relational Data-base Application -Focusing on Object-Oriented Analysis and Design Methodology-", The Korea Society of Computer and Information, Vol. 15, No.11, pp. 25-34, July 2011.