

애플릿을 이용한 웹 기반의 컴포넌트 뷰에 관한 연구

A Study of Web based Component View using Applet

임종혁, 정대성, 한정수
천안대학교

Jone-Hyuk, Im, Dae-Sung, Jung, Jung-Su, Han
Cheonan University

요약

CBD 개발 방법론의 발전으로 많은 컴포넌트가 개발되면서 소프트웨어의 빠른 기술변화와 더불어 상용컴포넌트 사용이 활성화되고 있다. 본 논문에서는 웹 상에서 컴포넌트의 클래스 다이어그램과 명세를 보여주는 뷰를 구현하였다. 이를 위해 각 정보를 DB에 분류 저장하여 애플릿을 이용한 뷰어에서 검색하여 사용할 수 있게 하였다.

Abstract

As a lot of components are developed by CBD methodology, COTS are activated along with fast technology change of software. In this paper, implemented a view that show class diagram and specification of component in web. For this, we classified components in database and each information is used to retrieve through the view.

I. 서론

소프트웨어 개발방법이 상용컴포넌트, 표준컴포넌트 등을 사용하면서 새로운 패러다임으로 진행되고 있다. 컴포넌트 개발 방법론은 기존보다 빠른 개발을 할 수 있는 장점을 가진다. 이것은 시스템 개발 시 기존처럼 하나에 시스템을 구성할 때 컴포넌트가 집약되어 새로운 시스템이 만들어짐을 의미한다. 최근엔 소프트웨어 형상관리(SCM : Software Configuration Management)가 부각되는데, 이것은 불안정한 구성요소를 가지는 시스템에서 변경이 일어나는 관리기술을 말한다. [1][2].

컴포넌트 기반 개발 방법을 사용하면서 UML의 사용이 빈번하게 대두되면서, 각 컴포넌트의 클래스 다이어그램, 시퀀스 다이어그램 등을 웹에서 능동적으로 보여줄 수 있고 필요에 따라서 해당 다이어그램을 다운로드 받을 수 있는 프로그램 개발을 유도하고자 한다.

본 논문에서는 자바 애플릿을 이용하여 웹 뷰어를 개발 하였고 MySQL을 이용하여 데이터베이스를 구축하였고 자바 JDBC를 이용하여 데이터베이스를 연동하였다.

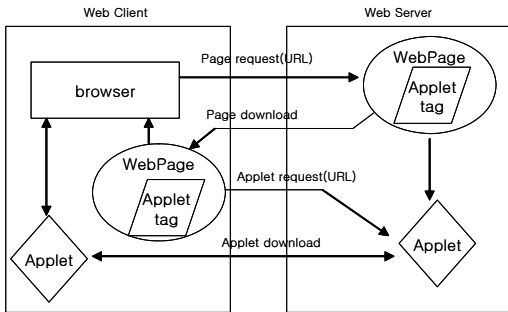
II. 관련연구

2.1 자바 애플릿

자바 애플릿은 HTML 페이지에 포함되어 자바 호환 웹브라우저에 의해 실행될 수 있는 자바 프로그램이기 때문에 자바 호환 웹 브라우저가 자바 애플릿이 포함된 HTML 페이지를 보여줄 때, 웹 서버 쪽에 있는 자바 애플릿 코드를 다운로드 한 후 브라우저 내의 특정 영역에서 실행하게 된다. 이렇게 자바 애플릿을 HTML 페이지에 삽입하기 위해서는 <applet> 태그를 사용한다. 이러한 자바 애플릿이 악의적으로 사용될 경우 사용자 시스템에 유해한 동작을 유발 시

킬 수 있다.[3]

그림 1.은 웹 상에서 애플릿이 동작하는 절차를 보여주고 있다. 사용자는 웹 브라우저를 통해 Web Server의 Web Page를 요청을 하게 되면 Client의 웹 브라우저에서는 server측의 Web Page를 애플릿 태그와 함께 다운로드 하여 웹 브라우저에 나타내 주게 된다. 이 때, Client측으로 다운로드 된 Web Page의 애플릿 태그는 서버 측의 애플릿을 다운로드 하여 Client의 웹 브라우저에 나타내 주게 된다.



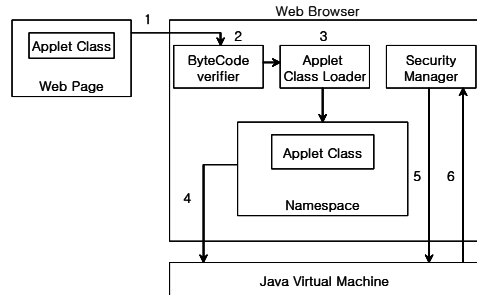
▶▶ 그림 1. 웹 상에서 애플릿 동작 절차 (Execution Process of Application The Web)

2.2 자바 애플릿 보안

자바 애플릿을 다운로드 받아 실행 시키기 위해서는 자바 applet viewer나 자바를 지원하는 웹 브라우저를 사용해야 한다. 자바 애플릿은 웹 브라우저에서 제공되는 sandbox에 의한 애플릿 실행 환경에서만 활동할 수 있다. 네트워크를 통해 전송된 애플릿은 신뢰성이 없는 것으로 간주하여 반드시 sandbox 내에 있는 보안에 따라 시스템 자원에 대한 접근을 제한한다.

그림 2.는 자바 애플릿의 보안 처리 과정을 보여주고 있다. Web Browser에서 애플릿 코드를 가져오면 ByteCode 검사기에서 애플릿 코드를 검사하고, 코드에 이름공간을 할당하여 JVM(Java Virtual Machine)을 이용하여 애플릿을 실행하게 된다. 이 때, 보안 매니저는 위험한 코드가 실행되면 코드검사를 하고, 애플릿의 일부 동작을 제한한다.

플릿의 일부 동작을 제한한다.

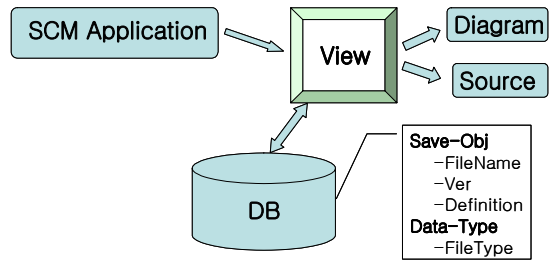


▶▶ 그림 2. 자바 애플릿의 보안 처리 과정 (Security Management Process of Java Applet)

III. 컴포넌트 뷰

3.1 시스템 구조

개발시스템 운영체제는 윈도우2000 플랫폼을 이용하였으며, UML 툴로는 투게더(Together 6.0)을 이용하였고 개발언어는 자바 1.3.1를 이용하였다.



▶▶ 그림 3. 시스템 구조

먼저 시스템에서 컴포넌트 접속시 어플리케이션에서 제공되는 뷰(View) 포인트를 통하여 해당 컴포넌트의 클래스 다이어그램과 소스(Source)를 보여줌으로써, 해당 컴포넌트 정보를 보다 상세하게 알 수 있다. 뷰는 컴포넌트 다이어그램과 소스를 보여주게 되며, 컴포넌트와 관련된 정보는 데이터베이스를 이용한다. 데이터베이스는 이와 관련된 저장된 객체이름과 객체 데이터형태 정보를 담고 있어야 한다. 저장

객체에는 파일이름과 버전, 정의 등이 저장되어야 하고, 데이터 타입에는 데이터형태가 저장된다.

3.2 뷰 구축

데이터베이스 테이블 내 각각 필드에서 파일명은 사용하기에 따라서 버전으로 사용할 수 있으며, 따로 버전을 주는 방식을 택해도 무관하다. 정의는 컴포넌트 이름(기능) 등을 수반하게 된다. 그림 4는 이를 기반으로 한 DB Table구조를 보여준다. 데이터타입에서 파일타입은 각 컴포넌트의 파일형태를 의미하며 파일 형식뿐만 아닌 개발된 언어를 포함할 수 있다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE Component [
  <!ELEMENT Component (Save_Obj,
Data_type) >
  <!ELEMENT Save_obj (File_Name, Ver,
Definition*)>
  <!ELEMENT File_name (#PCDATA)>
  <!ELEMENT Ver (#PCDATA)>
  <!ELEMENT Definition (#PCDATA)>
  <!ELEMENT Data_type (#PCDATA)>
  <!ELEMENT File_type (#PCDATA)>
]>
```

▶▶ 그림 4. DB 저장구조 DTD

```
try{
  Class.forName("org.gjt.mm.mysql.Driver").n
ewInstance();
}
catch(Exception e) {
  e.printStackTrace();
}
```

▶▶ 그림 5. JDBC Driver 로딩

사용된 데이터베이스는 MySQL으로 자바의 JDBC 를 사용하여 연동하였고, 그림 5는 MySQL의 JDBC

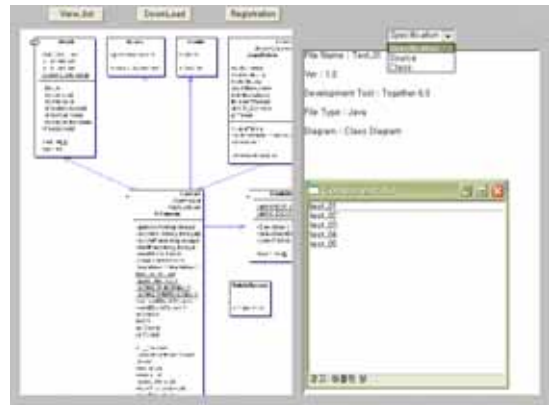
Driver를 로딩하기 위한 코드이다. 그림에서는 클래스 Class의 forName()메소드를 이용하여 드라이버를 직접 선택하여 로딩하고 있다. 자바의 JDBC 에서는 데이터베이스를 사용하기 위해 드라이버를 제공해 주고 있고, 이렇게 로딩된 데이터베이스는 그림 6.에서처럼 Connection 처리를 해주므로써 데이터베이스에 접근하여 자원을 이용할 수가 있게 된다. 그림에서는 지정된 User의 아이디와 패스워드로 Connection 객체를 생성 해주고 있다. 이렇게 생성된 Connection 객체를 통해서 한번 접속이 이루어지면 모든 작업들은 이미 생성된 객체를 계속해서 사용할 수 가 있다.

```
String dbURL=
"jdbc:mysql://localhost/java?user=root&passwo
rd=1012";

Connection conn =
DriverManager.getConnection(dbURL);

System.out.println("Connencted Database
Server");
```

▶▶ 그림 6. Connection



▶▶ 그림 7. 프로그램 구동

그림 7은 구현된 애플릿 화면을 보여주고 있다. 애플릿

플릿 화면을 통해 웹상에서 데이터베이스에 접근, 원하는 컴포넌트를 검색하여 사용할 수 있다. View_list는 DB에 저장된 컴포넌트 목록을 보여주고, Download는 원하는 컴포넌트를 다운을 받아 사용할 수 있도록 해주고 있고, Registration은 컴포넌트를 등록 기능을 수행한다. 컴포넌트를 선택했을 경우 Choice 버튼을 이용하여 Specification, Source, Class중에 하나를 선택하면, 관련 정보를 텍스트 필드 창에 나타내 준다.

IV. 결 론

본 논문에서는 컴포넌트 기반 개발방법을 보다 효율적으로 운용하기 위해 기존의 오프라인형 방식을 탈피하여 웹상에서, 컴포넌트 조립시 웹상에 등록된 컴포넌트를 보다 적절하고 쉽게 시스템에 적용시키기 위한 방법을 제안하였다.

여기서는 데이터베이스에 등록된 컴포넌트를 웹상에서 애플릿을 이용하여 컴포넌트 정보를 다이어그램으로 보여주고 이에 관한 명세를 함께 보여줌으로써, 컴포넌트 사용시 보다 정확한 정보를 전달하여, 시스템에 보다 적절한 컴포넌트를 선택하고 선택된 컴포넌트를 다운로드 받아 사용할 수 있도록 하였다.

■ 참고문헌 ■

- [1] Ivica Crnkovic, Peter Funk, Magnus Larsson, "Processing Requirements by Software Configuration Management", In euromicro 99, proceedings of the 25th EUROMICRO conference Milano, Italy, Sep 1999.
- [2] Magnus Larsson, Ivica Crnkovic, "Component Configuration Management", In ECOOP Conference, Workshop on Component Oriented Programming Nice, France, June 2000.
- [3] Li Gong, "Java Security Architecture(JDK1.2)", Sun Microsystems, 1997.