사용자 세션 지원 자바 서블릿

김 진 홍[†] · 정 현 락[††] · 박 양 수[†††] · 이 명 준[†††]

요 약

웹브라우저를 통하여 접속한 사용자들의 세션들을 관리하기 위해서는 웹서버는 사용자 인증과 사용자 정보 관리 기능이 필요하다. 이러한 기능은 웹서버가 사용자의 상태를 유지할 수 있게 하고, 이러한 사용자 상태에 따라 구별된 서비스를 제공할 수 있도록 한다. 현재 이러한 기능은 표준화되지 않은 기술을 이용하여 제공되고 있어, 많은 웹 프로그래머들은 유사한 기능의 개발을 반복적으로 하고 있다.

본 논문에서는 이러한 문제를 해결하기 위하여, 사용자에 대한 정보를 유지할 수 있는 새로운 세션 클래스에 대하여 정의하고, 사용자 인증, 세션관리, 그리고 사용자의 그룹 정보에 따라 요청 서비스를 제어하는 서블릿 클래스들에 대한 설계와 구현에 대하여 기술한다. 이러한 서블릿 클래스들은 JavaServer, Apache, 그리고 IIS와 같은 서블릿을 지원하는 웹서버에서 동작하게 된다.

Java Servlets Supporting Users' Sessions

Jin-Hong Kim[†] · Hyun-Rak Jung[††] · Yang-Su Park[†††] · Myung-Joon Lee[†††]

ABSTRACT

To maintain *users' sessions* through web browsers, a web server needs facilities for authenticating users and managing their information. Those facilities enable a web server to keep the states of the users, providing services distinguished by the states. As of now, those facilities are provided through non-standardized technologies, leading to the repeated development of similar functionalities through many web programmers.

In this paper, to solve the problem, we define a new *session class* containing the information on a user, and present a design and implementation of the *Java servlet classes* which supply the facilities for authenticating users, managing their sessions, and controlling web services according to the groups they belong to. These servlet classes work on any web servers supporting Java servlet such as JavaServer, Apache, and IIS.

1. 서 론

정보화 시대가 본격화되고 인터넷이 널리 보급되면서 웹브라우저를 이용하여 자신이 원하는 정보를 찾고 서비스를 요구하는 사용자가 날로 늘어나고 있다. 이러한 웹서비스에서 웹서버에 접속한 사용자 정보에 따라 특정 서비스를 제공하기 위하여 웹서버는 사용자 인증과 사용자 정보 관리 기능이 필요하다. 현재 널리 쓰이고 있는 웹서버들은 표준화되지 않은 서로 다른 기술을 이용하여 이러한 기능을 제공하고 있다. 따라서 사용자 인증[1, 2]과 사용자 정보[3] 관리 기능을 지원하는 웹서비스를 개발할 때마다 시스템의 환경에 따라 이러한 기능을 제공하기 위하여 유사한 작업을 반복적으로 하고 있는 형편이다.

본 논문에서는 이러한 문제점에 대한 해결책으로서, 분산환경에 적합하고, 이식성이 높으며, 코드 재사용이

※ 본 결과는 정보통신부의 정보통신 우수시범학교 지원사업에 의하여 수행된 것입니다.
† 정 회 원 : 울산대학교 대학원 정보통신공학부
†† 정 회 원 : 한국 오라클 CIP AHI실에 근무 중
††† 정 회 원 : 울산대학교 정보통신공학부 교수
논문접수 : 1999년 8월 9일, 심사완료 : 2000년 4월 24일

용이한 자바[4]와, 웹서버의 기능을 확장하기 위해 제공된 자바소프트의 서블릿 API[5, 6]를 사용하여 웹서버 종류에 관계없이 사용자 인증과 사용자 정보 관리 기능을 지원하도록 개발된 세션 서블릿[7]에 대하여 기술한다.

웹 서버에서 사용자 정보 관리 기능을 구현하는 방법으로 세션 객체를 이용하는 방법이 대표적이다. 세션이란 인증 과정을 거친 사용자 정보를 저장하는 객체로써, 인증 과정[8]을 거친 사용자는 자신의 정보를 가진 유일한 세션을 가진다. HTTP는 연속된 연결을 유지할 수가 없지만, 세션은 사용자 웹브라우저에 전달되어 유지될 수 있다. 그러므로 웹서버는 이러한 세션을 이용하여 사용자 관리 기능을 구현할 수 있고, 자신에게 접속한 사용자의 상태를 알아 볼 수 있다.

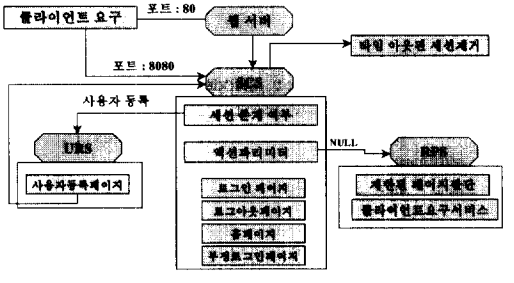
웹서버에서의 세션 관리 방법은 마이크로소프트사에서 제공하는 ASP(Active Server Pages)[9], 자바소프트에서 제공하는 JSDK(Java Servlet Development Kit)[10]를 이용하는 방법 등이 있다. ASP에서는 세션 컴포넌트를 이용하여 사용자가 웹서버에 접속하였을 때, 사용자에 대한 세션 객체를 생성하고 관리를 위한 API를 제공하고 있다. 그러나, 제공된 세션 컴포넌트의 API는 마이크로소프트사의 특정 웹서버인 IIS(Internet Information Server)[11]에서만 사용 가능하다. 그리고 JSDK는 웹서버 기능 확장을 위한 서블릿 클래스를 포함하고 있다. 이 서블릿 클래스 중에서 세션 클래스를 이용하여 사용자 정보를 유지할 수 있지만, 사용자 인증과 세션 관리를 위하여 사용자가 직접 관련된 프로그래밍 작업을 하여야 하는 문제점을 가지고 있다.

개발된 사용자 세션 관리 시스템은 자바 언어로 새롭게 정의한 세션 클래스와 세 개의 서블릿 즉, 사용자인증 서블릿, 세션관리 서블릿, 그리고 제한페이지서비스 서블릿으로 구성된다. 각각의 서블릿은 다음과 같은 기능을 제공한다. 세션관리 서블릿은 웹서버를 통하여 요청된 사용자 메시지를 분석하여 해당 서블릿을 호출할 수 있는 웹페이지를 사용자에게 보내는 기능과, 새롭게 인증된 사용자에 대하여 사용자 아이디를 기반으로 새로운 세션 객체를 생성하고 일정시간 동안 유지하는 기능을 한다. 사용자관리 서블릿은 처음 웹서버에 접속한 사용자에 대하여 사용자 등록 기능과 새로운 세션을 생성하기 위하여 사용자 인증기능을 수행한다. 제한페이지서비스 서블릿은 웹서비스를 요청하는 사용자 그룹 정보에 따라 웹페이지를 서비스하는 기능을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 세션관리를 위한 세션클래스 정의와 세션 관리 기법에 대하여 기술한다. 3장에서는 사용자 인증을 위한 사용자 관리에 대하여 설명하고, 4장에서는 사용자의 요구 사항을 사용자가 속한 그룹 정보에 따라 서비스를 제한하는 방법에 대하여 기술한다. 5장에서는 사용자 인증과 세션 관리 시스템의 동작을 설명하고, 6장에서는 세션관리 기술에 대한 특징과 성능을 분석하며, 마지막으로 7장에서 결론을 맺는다.

2. 사용자 서비스 및 세션 관리

웹서버에 접속한 사용자의 세션 관리 기능은 접속한 사용자의 정보에 따라 웹서버가 다른 서비스를 할 수 있도록 한다. 구현된 세션관리 서블릿(SCS-SessionControlServlet)은 사용자의 요청을 검사하여 요구된 서비스를 사용자에게 보내는 기능과 사용자의 세션 관리 기능을 한다. (그림 1)은 서블릿을 이용한 사용자의 세션관리 서블릿을 보여주고 있다.



(그림 1) 세션관리 서블릿

2.1 사용자 서비스

세션관리 서블릿은 사용자의 요구 메시지에 포함된 액션 파라미터와 사용자 세션객체의 유무에 따라 해당 서비스를 제공한다. 액션 파라미터는 showLogin, showInvalid, login, logout 또는 널(null)이 될 수 있다. 사용자 메시지에서 사용자 세션객체가 없을 때는 액션파라미터에 showLogin값을 넣어서 다시 세션관리 서블릿을 호출하여 사용자 인증과정을 거치도록 한다. showInvalid 값은 사용자 인증에서 문제가 생겼을 때 사용자에게 보내주기 위해 사용된다. 파라미터에 login과 logout값이 들어왔을 때 해당 서비스를 세션관리 서블릿에서 제공한다.

2.2 세션 관리

세션관리 기능은 웹서버에 접속한 사용자 각각에 대한 세션 객체를 생성, 유지, 소멸할 수 있는 기능이다. 세션관리 서블릿은 새롭게 접속한 사용자의 세션객체를 생성하고, 이를 쿠키(Cookie)를 이용하여 사용자 웹브라우저에게 전달됨과 동시에 자신의 세션캐쉬에 생성된 세션객체를 유지한다. 이러한 사용자 웹브라우저에서 유지하는 세션객체와 세션관리 서블릿에서 관리하는 세션객체를 이용하여, 세션관리 서블릿은 웹서버에 접속한 사용자중에서 유효한 사용자를 판단하여 사용자 정보에 따른 서비스를 한다. 이를 위하여, 세션관리 서블릿은 세션 클래스를 이용하여 새로운 세션 객체를 만들어내는 기능과 생성된 세션을 관리하는 기능을 가진다.

2.2.1 세션 클래스

세션은 웹서버에 접속한 사용자마다 하나씩 생성되어 웹서버가 접속한 사용자 정보를 유지할 수 있도록 한다. 이러한 세션은 접속한 사용자가 인증의 과정을 거치고 나서 사용자 아이디에 따라 각각의 사용자에게 유일하게 할당된다. (그림 2)는 세션 클래스를 보여 주고 있다.

```
Session (String userid) {
    _userId = URLEncoder.encode(userId);
    _id = Math.abs(newRandom
          (System.currentTimeMillis()).nextInt());
    _expires = 0;
    _uri = new String();
}
```

(그림 2) 세션 클래스

세션 클래스는 사용자 정보를 저장하는 속성 값과 이 값을 조작할 수 있는 함수로 구성 되어있다. 이러한 속성 값은 사용자의 정보를 보관하는 것으로써 사용자 식별자, 세션 식별자, 활동 시간, 그리고 URI로 구성되어 있다.

사용자 식별자는 사용자 인증 과정에서의 사용자 아이디와 같은 정보를 가진다. 각 사용자마다 유일한 아이디를 가지므로 이 속성 값은 사용자 세션마다 유일한 값이 유지되며 어떤 사용자가 접속하고 있는지를 파악할 때 사용된다. 세션 식별자는 세션이 만들어지는 시각을 이용하여 만들어지는 값으로써 서버에 존재하는 각각의 세션은 유일한 값을 가진다. 그러므로 세션관리 서블릿은 이 속성 값을 통하여 서버에 존재하는 세션을 구별하는데 사용한다. 활동 시간은 특정 사용자에게 만들어진 세션이 사용자로부터 요청 메시지나 응답이 없을 때, 세션관리 서블릿의 세션캐쉬에 세션이 저장되는 기간을 나타낸다. 이 속성 값을 통하여 세션관리 서블릿은 활동 시간이내에 사용자로부터 응답이나 요구사항이 없을 때, 자동적으로 해당 사용자 세션을 세션캐쉬에서 삭제한다. URI속성은 현재 사용자가 어떤 주소 페이지에 있는지를 알려주기 위한 속성이다.

2.2.2 세션 생성 및 정보 유지

새로운 세션객체가 세션관리 서블릿에 의하여 생성되는 시기는 인증 과정을 거친 사용자의 요청 메시지에 세션 객체가 존재하지 않은 시점이다. 접속한 사용자에 대한 새로운 세션 생성 과정은 다음과 같다.

세션관리 서블릿은 사용자 요청 메시지에서 쿠키 정보를 조사하여 사용자에 대한 유효한 세션정보를 포함하고 있는지를 알아본다. 유효한 세션객체를 가지고 있지 않을 경우, 사용자 인증 과정을 거치도록 하여 그 사용자의 식별자를 기반으로 새로운 세션을 만들어낸다. 이렇게 생성된 세션객체는 사용자 식별자, 세션 식별자, 활동 시간, 그리고 URI에 대한 정보를 포함하고 있다. 사용자에 대한 새롭게 생성된 세션객체는 세션관리 서블릿 내에 존재하는 세션캐쉬(SessionCache) 저장됨과 동시에 쿠키를 이용하여 클라이언트에게 보내진다. 그리하여 세션관리 서블릿과 사용자 웹브라우저는 같은 사용자 식별자를 가지는 유효한 세션을 유지한다. (그림 3)에서는 새롭게 만들어진 세션을 세션관리 서블릿의 세션캐쉬에 저장하고, 새로운 쿠키를 생성하여 세션 정보를 포함시켜 사용자 웹브라우저에게 재 전송하는 것을 보여주고 있다.

```
session = new Session (userId);
session.setExpires (sessionTimeout+System.currentTimeMillis());
sessionCache.put (session.key(), session);

Cookie c = new Cookie("SessionMnager", String.valueOf (session.getId()));
c.setPath ("/");
c.setMaxAge (-1);
c.setDomain ("serverIP);
response.addCookie(c);
```

(그림 3) 새로운 세션 만들기

반면, 세션관리 서블릿은 이미 인증 과정을 거친 사용자 요청에 유효한 세션을 포함하고 있다면, 사용자 인증 과정 필요 없이 바로 요청한 서비스를 제공한다. 이러한 과정을 위하여 세션관리 서블릿은 유효한 세션을 판단하는 기능을 할 수 있다. 유효한 세션 판단은 다음절에서 자세히 설명한다.

2.2.3 유효한 세션 판단

사용자 인증을 거친 사용자의 웹브라우저는 웹서버에게 세션객체 정보를 포함한 서비스 요청 메시지를 보낸다. 이러한 세션객체는 세션 식별자, 사용자 식별자 등 속성 값을 가지고 있다. 또한 세션관리 서블릿의 세션캐쉬에 저장된 세션들도 자신의 유일한 세션 식별자를 갖는다. 그러므로 유효한 세션 판단은 두 개의 세션 즉, 사용자 웹브라우저가 보낸 세션과 세션관리 서블릿이 관리하고 있는 세션이 가지고 있는 세션 식별자를 비교하여 이루어진다. 두 개 세션의 세션 식별자가 일치한다면 세션관리 서블릿은 사용자 웹브라우저가 유효한 세션을 가지고 있다고 판단하여 요청한 서비스를 제공한다. 반대로 일치하지 않다면, 세션관리 서블릿은 사용자 인증을 위한 페이지를 사용자에게 전송하여 새로운 인증과정을 통한 새로운 세션을 만들도록 한다.

유효한 세션을 관리하는 세션관리 서블릿은 일정 시간동안 사용자의 서비스요구가 없을 시에는 저장된 세션 정보를 삭제하도록 한다. (그림 4)는 사용자 웹브라우저를 통하여 보내진 사용자 요청 메시지에서 쿠키정보를 얻어내고, 이 쿠키 정보에서 세션 식별자를 알아낸다. 이는 알아낸 세션 식별자를 이용하여 웹서버 측에서 동일한 세션이 있는지를 알아내는 과정을 보여주고 있다.

```
Cookie c[] = request.getCookies();
        Session session;
if (c != null) {
        String key = String.valueOf (c[0].getValue());
        session = (Session)sessionCache.get (key);
}else {
        System.out.println("No Cookie  in Current repuest");
        session = null;
}
```

(그림 4) 세션 관리

2.2.4 세션의 소멸

세션관리 서블릿이 받은 메시지의 액션 파라미터에 logout값을 가진 경우와 세션관리 서블릿이 저장된 각 세션들의 활동 시간을 체크하여 활동 시간을 초과한 세션이 있을 경우, 세션관리 서블릿은 세션캐쉬에서 세션을 제거하여 소멸시킨다. (그림 5)는 세션관리 서블릿에서 일정간격으로 세션캐쉬에 저장된 세션의 활동 시간을 파악을 하여 활동 시간을 초과한 세션을 삭제하는 기능을 보여주고 있다

```
synchronized (sessionCache) {
    sessions = sessionCache.elements();
}
while ( sessions.hasMoreElements() ) {
    s = (Session)sessions.nextElement();
    if ( expire >= s.getExpires() ) {
        synchronized (sessionCache) {
            sessionCache.remove (s.key());
        }
    }
}
```

(그림 5) 세션 관리 - 삭제

3. 사용자 관리

사용자 인증 기능을 위해서 웹서버는 새로운 사용자에 대한 등록기능, 등록한 사용자 정보 관리 기능, 그리고 사용자 로그인 기능이 필요하다. 본 논문에서는 사용자에 대한 정보를 새로 정의한 사용자 클래스(User class)의 형태로 파일 시스템을 이용하여 유지한다. 그리고 사용자관리 서블릿은 사용자 인증을 위한 기능을 제공한다.

3.1 사용자 클래스

사용자의 정보를 나타내는 사용자 클래스는 (그림 6)과 같이 Id, Name, Passwd, 사용자의 소속 그룹을 나타내는 Group등의 속성을 가지고 있다.

```
class User {
        String Id;
        String Name;
        String Passwd;
        String Group;
}
```

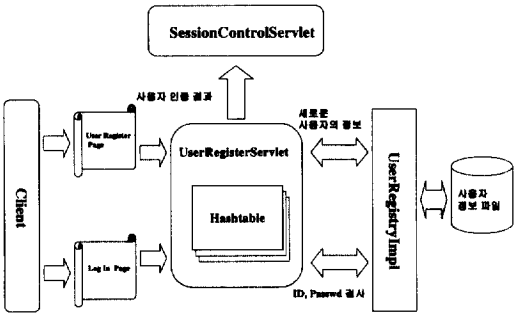
(그림 6) 사용자 클래스

사용자 클래스 속성 값은 새로운 사용자가 처음 웹서버에 접속하여 자신의 정보를 등록하는 과정에서 값

이 정해진다. 이 클래스 속성 값 중에서, 사용자 아이디는 접속한 사용자의 새로운 세션을 생성될 때 사용될 것이며, 사용자 아이디와 패스워드는 사용자 인증 과정에서 필요한 값이다.

3.2 사용자인증 서블릿

사용자인증 서블릿(URS-UserRegistryServlet)은 사용자 등록 및 인증을 담당하는 서블릿으로 웹서버 측에서 운용되는 서블릿이다. 이 서블릿은 사용자 정보를 사용자 정보 파일에 읽고 쓰기를 하기 위하여 사용자 등록실행(UserRegistryImpl) 클래스를 이용하고 있다. 사용자 인증 및 등록을 위한 구조는 (그림 7)과 같다.



(그림 7) 사용자 등록 및 인증 과정

세션관리 서블릿은 사용자의 요청 메시지를 분석하여 세션의 유무를 가려낸다. 접속한 사용자의 웹브라우저가 유효한 세션을 가지고 있다면 요청된 서비스를 제공하지만, 가지고 있지 않을 경우, 새로운 사용자인지 기존 사용자인지에 따라 다른 HTML 페이지를 사용자에게 제공한다. 세션관리 서블릿은 접속한 사용자가 새로운 사용자인 경우 사용자 등록을 할 수 있는 HTML 페이지를, 기존 사용자인 경우 인증과정을 위한 HTML 페이지를 사용자에게 보낸다. 사용자 등록을 위한 HTML 페이지는 사용자 정보를 기입할 수 있는 필드를 가지고 있으며, 사용자 인증을 위한 HTML 페이지는 사용자 아이디와 패스워드를 기입할 수 있는 필드를 가지고 있다. 두 HTML 페이지는 각 필드 값을 인자로 하여 사용자인증 서블릿을 호출할 수 있는 링크를 가지고 있다. 이 링크를 통하여 사용자인증 서블릿을 호출하여 사용자 인증 및 사용자 등록을 할 수 있다.

사용자인증 서블릿은 사용자 브라우저로부터 받은 정보와 기존의 사용자정보를 관리하는 사용자등록실행 클래스로부터 얻은 정보를 비교하여 사용자 인증과 사용자 등록을 수행을 한다. 이 수행 결과는 다시 세션관리 서블릿에 보내어 사용자에게 보낸다.

3.3 사용자등록실행 클래스

사용자등록실행 클래스는 사용자등록 서블릿에 의해 전달된 새로운 사용자의 정보를 파일에 저장하는 기능과 기존 사용자의 정보를 읽어오는 기능 및 Multi-Read-Single-Write를 지원하기 위한 동시성 제어를[12] 담당한다. 즉, 사용자등록실행 클래스는 사용자의 정보를 다루는 실제적인 기능을 한다. 구현된 사용자등록실행 클래스는 사용자 정보를 유지를 위하여 파일시스템을 이용한다. 이 방법은 사용자 아이디, 사용자 이름, 패스워드, 그룹에 대한 정보 등 적은 용량의 사용자 정보를 저장을 위하여 기존 데이터베이스를 설치해야 하는 부담을 줄이기 위한 것이다. (그림 8)은 사용자 정보를 지정된 파일에 쓰는 과정을 보여주고 있다. (그림 8)에서의 setUser() 함수는 새로 등록한 사용자 정보가 이미 존재하는 사용자 정보인지를 사용자 아이디를 이용하여 파악한다. 그리고 중복되는 사용자 아이디가 파일에 존재하지 않을 때, 새로 등록한 사용자 정보를 지정된 파일에 저장한다. 이때 다중 사용자들이 동시에 지정된 파일에 접근하여 올바르게 데이터 변경을 하기 위하여 파일에 대한 동시성을 고려하여야 한다. 파일에

```
public boolean setUser (String dir, User member)throws AdminException {
  se.lock();
  User m = (User)_table.get(member.getUserId());
  if (m != null) return false;
  else {
    _table.put(member.getUserId(), member);
    File file = new File(dir, "member.dat");
    try{
      ObjectOutputStream out = new  ObjectOutputStream(
      new FileOutputStream(file));
      out.writeObject(_table);
      out.flush();
      out.close();
    }catch(Exception e){
      e.printStackTrace();
    }
    return true;
  }
  se.unlock();
}
```

(그림 8) 사용자 정보 관리

대한 동시성을 위하여 setUser()함수는 세마포어 객체의 함수(lock(), unlock())를 이용하여 다중 사용자의 사용자 정보 파일에 대한 접근을 제어하고 있다.

4. 제한된 페이지 서비스 관리

세션관리 서블릿에 의하여 웹서버는 접속한 사용자 정보를 일정시간 동안 유지할 수 있다. 그리하여 웹서버는 접속한 사용자 정보에 따라 서로 다른 서비스를 사용자에게 제공할 수 있다. 구현된 제한페이지서비스 서블릿(RPSS-RestrictedPagesServiceServlet)은 사용자 그룹 정보를 가진 웹페이지를 등록하는 기능과 인증된 사용자의 세션객체에서 얻어낸 사용자 그룹 정보를 이용하여 사용자 그룹별로 서비스를 할 수 있는 기능을 제공한다.

4.1 제한된 페이지 관리

제한페이지서비스 서블릿은 접속한 사용자의 세션 정보 중에서 사용자 그룹정보에 따라 서비스할 웹페이지를 관리한다. 사용자에 따라 서비스될 페이지 정보는 파일시스템을 이용하여 저장된다. 저장되는 웹페이지에 대한 정보는 웹페이지 파일 이름을 포함한 경로와 사용자 그룹정보이다. 파일 이름을 포함한 경로는 서비스될 페이지 위치를 나타내고, 사용자 그룹정보는 접속한 사용자가 속한 그룹정보와 비교되어 웹페이지가 어떤 사용자에게 서비스될지를 나타낸다.

제한페이지서비스 서블릿은 제한페이지에 대한 정보를 (그림 9)의 페이지정보(PageInfo) 객체로 관리된다. 페이지정보 클래스는 파일에 저장된 제한페이지 정보를 유지할 수 있는 속성을 가지고 있으며, 이러한 속성 값을 지정하고 읽어 오는 함수들로 구성되어 있다.

```
class PageInfo {
        String PagePathName;
        String Group;
}
```

(그림 9) 제한 페이지 클래스

4.2 제한된 페이지에 대한 그룹별 서비스

제한페이지서비스 서블릿은 사용자가 요청한 웹페이지의 접근 권한을 가지고 있는지를 결정하여, 그 결과를 세션관리 서블릿에게 전달하여 세션관리 서블릿이 요구한 서비스를 제공할 수 있게 한다. 이 서블릿은 특정 웹페이지에 대한 접근 허용을 사용자가 속한 그룹의 정보에 따라 결정하기 위하여, 사용자로부터 접근이 제어되어야할 웹페이지 리스트를 가지고 있고, 서비스를 요청한 사용자의 그룹 정보와 요청된 웹페이지 그룹 정보와 비교하는 기능을 가진다.

사용자로부터 접근이 제어되어야할 웹페이지 정보는 4.1절에 기술한 페이지정보 클래스를 이용하여 제한페이지서비스 서블릿에서 관리된다. 이 서블릿이 처음 기동될 때, 웹서버에서 관리될 모든 제한된 웹페이지에 대한 정보를 가진 파일에서 웹페이지 정보를 읽어서 각 페이지정보 객체를 생성한다. 생성된 페이지정보 객체는 페이지캐쉬에 저장되어 사용자로부터 요청된 웹페이지를 검증하는데 사용된다.

(그림 10)에서 사용자의 그룹 정보에 따라 제한된 페이지를 서비스하는 것을 보여주고 있다.

```
public class RestrictedSessionServlet extends SessionManager {
  protected void serviceRequest (Session session,
  HttpServletRequest request, HttpServletResponse
     response) throws IOException, ServletException {
  boolean allowed = true;
  String pageGroup = checkPage(request.getRequestURI());
  if( pageGroup != null) {
     if( session.getUserGroup().equals(pageGroup) ) {
        allowed = true;
     } else {
        allowed = false;
     }
  } else {
     allowed = true;
  }
  if( allowed ) {
     super.serviceRequest(session, request, response);
  } else {
     response.sendError(HttpServletResponse.SC_FORBIDDEN,"This
     page is restricted");
  }
}
```

(그림 10) 제한페이지 서비스

5. 사용자 및 세션 관리 시스템

5.1 전체시스템 구조

사용자 인증 기능과 세션을 관리해주는 시스템은 웹서버와 세션관리 서블릿, 사용자인증 서블릿, 그리고 제한된 페이지를 관리하여 서비스를 요청한 사용자의

정보에 따라 서비스하는 서블릿으로 구성된다.

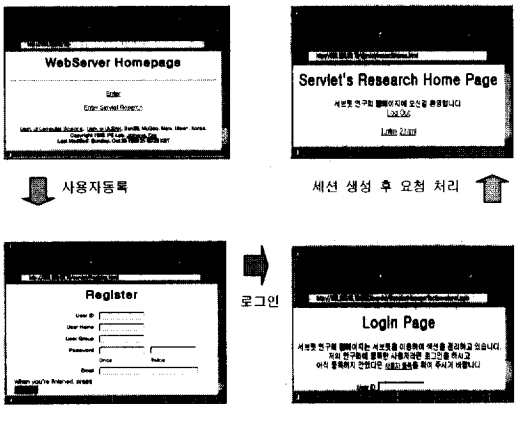
일반적인 사용자 요구사항은 웹서버가 받아서 처리를 한다. 그러나 제한된 페이지를 요청할 때, 웹서버는 직접 처리하지 않고 이 요청 서비스를 사용자 인증 과정을 거쳐 유효한 세션 정보를 가진 사용자에게만 서비스를 하는 세션관리 서블릿에게 처리하도록 한다. 세션관리 서블릿은 사용자가 보낸 요청 메시지에서 쿠키 정보를 조사하여 유효한 세션을 가지고 있는지를 알아낸다. 그리고 세션관리 서블릿은 쿠키 정보에서 유효한 세션을 가지고 있는지 않은지에 따라 두 방향의 서비스를 제공한다.

먼저, 사용자 요청 메시지의 쿠키 정보에 세션에 대한 정보를 가지고 있지 않거나, 있더라도 존재하는 세션정보가 유효하지 않을 경우, 세션관리 서블릿은 사용자가 웹서비스를 받기 이전 유효한 세션 정보를 얻기 위한 사용자 인증 과정을 거치도록 한다. 인증 후, 세션관리 서블릿은 새로운 세션을 사용자에게 쿠키를 이용하여 보내고, 자신 또한 동일한 세션정보를 유지한다.

반면, 사용자의 요구 메시지의 쿠키 정보에 유효한 세션정보를 가지는 경우, 세션관리 서블릿은 사용자로부터 보낸 메시지의 쿠키 정보에서 얻어낸 세션정보가 유효한 세션인지를 알아본다. 유효한 세션임을 확인 후, 제한된 페이지에 대한 서비스를 하도록 세션관리 서블릿은 제한페이지서비스 서블릿을 작동하게 한다.

(그림 11)은 사용자 인증과정과 사용자의 요구 사항을 서비스하는 것을 보여주고 있다.

세션관리 서블릿은 인증을 거친 모든 사용자의 세션



(그림 11) 클라이언트 요청 처리

정보를 유지하며, 웹서버를 로그 아웃한 사용자 세션을 자신의 Hash 테이블에서 삭제, 일정시간 접속한 사용자로부터 요청이 없을 때는 사용자의 세션 정보를 삭제하는 등의 세션 관리를 한다. 또한, 세션관리 서블릿은 사용자의 올바른 세션정보를 확인 후 사용자가 요청한 서비스를 처리하는 역할을 한다.

사용자인증 서블릿은 새로운 사용자 등록과 기존 사용자에 대한 인증 기능을 수행하며. 두 경우 모두 세션관리 서블릿이 사용자에게 제공한 HTML 문서에서 서블릿을 호출하는 태그를 이용하여 사용자인증 서블릿을 호출한다. 이 서블릿은 제공된 HTML 문서로부터 사용자 정보를 받아서 파일로 저장한다. 파일에 저장된 사용자 정보는 사용자 아이디를 사용하여 유일한 값을 유지하고, 이렇게 유지된 정보는 등록한 사용자가 인증의 과정을 거쳐 새로운 세션을 만들 때 사용된다. 즉, 인증의 과정에서 사용자의 정보를 파일로부터 얻어서 다시 세션관리 서블릿에게 이 정보를 전달한다. 그리하여 세션관리 서블릿은 해당 사용자에 대한 새로운 세션을 만들어 내어 사용자에게 서비스한다.

세션관리 서블릿에 의하여 호출되는 제한페이지서비스 서블릿은 특정 그룹에 속한 사용자에게 제공될 페이지 정보를 가지고 있다. 제한페이지서비스 서블릿은 이 정보와 그룹정보를 비교하여 사용자가 속한 그룹에 따라 특정 사용자에게 서비스를 제공한다.

5.2 실행 환경

WindowNT4.0과 JRun[15] 웹서버와 자바웹서버[16] 기반으로 세션관리 클래스와 사용자등록 클래스를 이용하여 세션관리 서블릿, 사용자등록 서블릿, 그리고 제한페이지 서블릿을 구현하였다. 구현된 서블릿을 이용하여 사용자 인증 및 세션관리 기능을 제공하기 위하여 웹서버는 이러한 서블릿을 등록한다. 서블릿을 등록하는 정보는 서블릿의 위치, 이름, 그리고 기동 시 필요한 파라미터들이다. 웹서버에 등록된 서블릿은 자신을 요청하는 서비스가 들어왔을 때 기동되어 사용자에게 서비스를 제공한다. 사용자측에서는 웹브라우저를 이용하여 쿠키를 포함한 서비스 요청을 웹서버에게 전달한다. 쿠키를 이용하여 사용자 세션 정보를 포함하는 사용자의 서비스 요청은 서블릿에서 분석하여 사용자 인증 및 세션관리를 한다.

시스템을 구현하는 도구로는 Borland JBuilder2.0을 사용하였으며, 서블릿의 작동 검사는 JSDK2.0에 포함

된 서블릿 실행 도구(servletrunner)를 사용하였다.

6. 관련 연구

웹서버 기능을 확장을 하기 위한 기술에는 마이크로소프트사에서 제공하는 방법인 ASP, 자바소프트에서 제공하는 방법인 자바 서블릿 등이 있다. 이러한 기술을 이용하여 웹서버는 사용자 정보를 관리하기 위한 세션 관리에 대한 방법을 제공하고 있다.

6.1 ASP를 이용한 세션 관리 방법

ASP를 이용한 세션 관리는 ASP에서 지원하는 컴포넌트중의 하나인 세션 컴포넌트를 사용한다[13]. 사용자가 웹서버에 접속하였을 때, 서버에서 제공하는 사용자 식별자를 이용하여 생성되는 ASP의 세션 객체는 사용자 정보를 유지하며 ASP의 어플리케이션에서 관리된다. 사용자의 세션 객체가 생성된 이후, 사용자가 스크립트와 HTML코드로 구성된 ASP 형식의 파일을 요청하였을 때 이 페이지는 사용자에게 할당된 세션 객체의 함수를 호출하여 서비스를 제공한다. 세션 객체에서 제공하는 함수에는 세션의 시작과 소멸할 때 호출되는 함수들과 세션정보를 반환하는 함수로 이루어져 있다. 이러한 ASP를 이용한 세션관리는 세션 객체의 함수를 이용하여 이루어진다. 그러나, ASP는 마이크로소프트사의 특정 웹서버인 IIS에서만 사용 가능하여 플랫폼 독립적인 웹서버 기능 확장에는 어려움이 있다.

6.2 자바 서블릿을 이용한 세션 관리 방법

자바 서블릿을 이용한 세션 관리는 자바 서블릿 API의 세션 클래스를 이용한다. 자바웹서버의 세션 트랙킹(Session Tracking)[14]은 접속한 사용마다 세션객체를 만들어 사용자의 상태를 유지할 수 있도록 하고 있다. 처음 웹서버에 접속한 사용자는 새로운 세션 객체와 유일한 세션 식별자를 가지게 된다. 사용자의 요청을 처리하는 서블릿은 사용자의 요청 메시지에서 세션 객체를 넘겨받아 이 객체의 정보를 수정하거나 유지할 수 있다. 그러나, 자바 서블릿 API의 세션 클래스를 이용하여 만들어진 세션 객체는 사용자 식별자와 사용자 이름 등의 사용자 정보를 포함하지 않는다. 이러한 사용자 정보를 가진 세션을 만들기 위해서는 사용자 인증을 하는 서블릿과 사용자 정보를 유지하는 새로운 세션 클래스를 필요시마다 제작하여야 한다.

본 논문에서 기술된 세션 관리 및 사용자 인증 서블릿은 자바 서블릿을 지원하는 웹서버에서 플랫폼 독립적으로 사용자 인증 및 세션 관리 기능을 제공한다. 이러한 서블릿들은 자바 서블릿 API를 이용하여 사용자의 요구사항을 처리하고, 사용자의 정보를 유지하기 위해서 새롭게 세션 클래스를 정의하였다.

7. 결론 및 추후 연구 과제

본 논문에서는 플랫폼 독립적인 사용자 인증과 세션 관리를 위한 세션 클래스와 자바 서블릿 클래스에 대하여 기술하였다. 이를 통하여 서블릿을 지원하는 웹서버 측에서 사용자 인증, 세션 관리, 그리고 사용자간의 정보에 따른 서비스 기능을 쉽게 구현할 수 있도록 하였다.

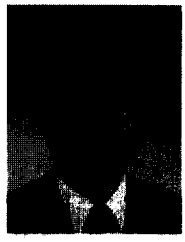
웹서버에서 사용자 인증과 세션 관리 기능을 확장하기 위하여, 자바를 이용하여 사용자의 정보를 저장할 수 있는 세션 클래스를 정의하였으며, JSDK2.0에서 제공하는 기본 서블릿 API를 이용하여 세 부분의 서블릿 클래스로 구현하였다. 각 사용자의 세션 관리와 사용자로부터 전달된 메시지를 분석하여 서비스하는 세션관리 서블릿이 개발되었으며, 또한 새로운 사용자 등록과 사용자 인증 기능을 수행하는 사용자관리 서블릿이 개발되었다. 사용자관리 서블릿은 새로운 사용자 정보를 받는 즉시 이 정보를 파일시스템을 이용하여 저장하도록 개발되었으며, 이를 통하여 추가적인 데이터베이스 시스템을 설치하는 부담이 해소되게 되었다. 마지막으로 인증 과정을 거친 각각의 사용자 정보 중, 그룹 정보에 따라 특정 서비스를 제공하는 제한페이지 서비스 서블릿을 구현하였다.

현재, 개발한 사용자관리 및 세션관리 서블릿을 이용하여 인터넷 환경에서 공동작업환경을 개발중이며, 또한 웹 개발자가 사용자 정보 관리를 위하여 특정 데이터베이스를 사용하고자 할 경우를 위하여 사용자관리 서블릿과 이 데이터베이스와의 연결을 지원하는 인터페이스를 개발하고 있다.

참 고 문 헌

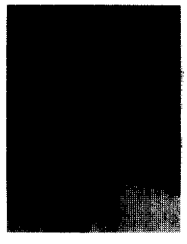
[1] Cem Paya, "A Framework for WWW client authentication protocols," March 1998.

[2] T. Berners-Lee, R. Fielding, H. Frystyk. RFC 1945 : "Hypertext Transfer Protocol HTTP/1.0," May 1996.
[3] D. Kristol, L. Montulli. RFC 2109 : "HTTP State Management Mechanism," February 1997.
[4] Sun MicroSystem Inc., "Java Language an Overview," http://java.sun.com/docs/white-/index.html#java-overview.ps
[5] Sun MicroSystems Inc., "THE JAVA SERVLET API," 1998.
[6] Sun MicroSystems Inc., "Servlet API Class Reference," http://java.sun.com/products-/servlet/2.1/html/api-reference.fm.html#3339, 1999.
[7] 김진홍, 정현락, 안건태, 한천용, 이명준, "사용자 세션을 관리하는 자바 서블릿", '99 한국정보과학회 봄 학술발표논문집 제26권, 1호, pp.379-381, 1999.
[8] Gene McKenna, "Combining Sessions and Cookies in an Authentication System", VP Product Development, BLUE-DOT.COM, December, 1998.
[9] Michael P. Levy, "ASP and Web Session Management," April 2, 1997.
[10] James Duncan Davidson, Sun MicroSystems Inc., "Java Servelt API Specification," November, 1998.
[11] Arthur Knowles, "Micorsoft Internet Information Server 3 Second Edition," Sms.net, 1997.
[12] Scott Oaks, Henry Wong, "JAVA threads," O' REILLEY, January 1997.
[13] Live Software Inc., "JRun Data Sheet", http://jrun.com/products/jrun/datasheet.htm-l#Documentation. 1998.
[14] Sun MicroSystems Inc., "Java Web Server," http://www.sun.com/software/jwebse-rver/overview/index.html, 1999.
[15] Alex Homer, et al. "Professional Active Server Pages," Wrox PressLtd, 1997.
[16] Sun Microsystems, Inc., "Java Web Server : Administrator Document-Session Tracking," http://jserv.java.sun.com/products/java-server/documentation/webserver1.1-/session_track/SessionTr.html, 1997.



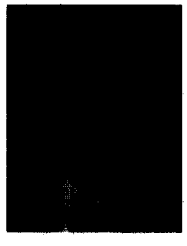
김 진 홍

e-mail : avenue@cic.ulsan.ac.kr
1999년 울산대학교 전자계산학과 졸업(학사)
1999년~현재 울산대학교 정보통신 공학부 공학석사과정
관심분야 : 웹 프로그래밍 시스템, 분산 컴퓨팅(이동 에이전트 시스템) 등



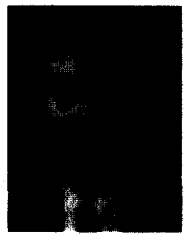
정 현 락

e-mail : hrjung@kr.oracle.com
1999년 울산대학교 전자계산학과 졸업(학사)
1999년 울산대학교 정보통신 공학부 공학석사과정 중 휴학
1999년 현재 한국 오라클 CIP AHI실에 근무 중
관심분야 : 프로그래밍 언어, 분산 객체 프로그래밍 등



박 양 수

e-mail : yspk@uou.ulsan.ac.kr
1978년 울산대학교 전자계산학과 졸업(학사)
1981년 서울대학교 계산통계학과 졸업(석사)
1986년 현재 서울대학교 계산통계학과 박사과정 수료
1980년 현재 울산대학교 정보통신 공학부 부교수
관심분야 : 분산처리, 컴퓨터알고리즘 등



이 명 준

e-mail : mjlee@uou.ulsan.ac.kr
1980년 서울대학교 수학과 졸업(학사)
1982년 한국과학기술원 전산학과 졸업(석사)
1991년 한국과학기술원 전산학과 졸업(박사)
1982년 현재 울산대학교 정보통신 공학부 교수
1993년~1994년 미국 버지니아대학 교환교수
관심분야 : 프로그래밍언어, 분산 객체 프로그래밍 시스템, 병행 실시간 컴퓨팅, 인터넷 프로그래밍시스템 등