

포탈 서버 시스템의 설계 및 구현

이형동^o 박철만 김낙현 김형주
서울대학교 컴퓨터공학부
{hdlee^o, cmpark, nhkim, hjk}@oopsia.snu.ac.kr

Design and Implementation of Portal Server System

Hyung-Dong Lee^o Cheol-Man Park Nak-Hyun Kim Hyoung-Joo Kim
Dept. of Computer Engineering, Seoul National University

요 약

최근 기업의 컴퓨팅 환경은 기존의 시스템 및 자료에 대한 통합화 개념으로 발전되고 있다. 포탈 서버는 이중의 응용 프로그램 및 데이터들을 통합하여 사용자에게 일관된 사용자 인터페이스를 통해 제공한다. 사용자는 개인화 서비스를 통해 쉽게 정보에 접근할 수 있으며, 타 사용자와의 정보 공유, 공동 작업 등을 통해 업무 효율을 높일 수 있다. 본 연구에서는 XML 기반의 포탈 서버 시스템을 설계하고 구현하였다. 이는 XML 응용 명세를 통해 정보 및 응용 프로그램에 대한 인터페이스를 제시하고, 포털릿을 통해 개인화된 사용자 인터페이스를 제공한다.

1. 서 론

포탈 서버는 포탈(portal)의 개념에서 기인되며, 포탈은 다양한 자료 소스(data sources)들에 대해 개인화(personalization)를 통한 일관된 뷰(view)를 제공하며 사용자에게 이에 대한 단일 접근점(single gateway)으로써의 역할을 하는 것이며, 포탈을 지원하기 위한 응용 시스템을 포탈 서버라 할 수 있다. 이러한 개념은 기존의 Yahoo!, Lycos 등의 웹사이트에서 출발한다. 이러한 웹사이트들은 웹(web)이라는 방대한 정보공간에서 사용자로 하여금 보다 쉽게 원하는 자료를 찾을 수 있도록 다양한 정보를 분류 저장하여 제공한다. 사용자는 해당 웹사이트를 시작점으로하여 여러 곳에 분산되어 있는 정보를 얻을 수 있게 된다. 이와 같이 웹 포탈은 분류된 자료체계와 검색 기능을 제공하여 사용자로 하여금 쉽게 웹 자원을 얻을 수 있게 해줌으로써 웹을 통한 정보 검색의 시작점으로써의 역할을 수행한다. 또한 최근의 웹 포탈 경향으로 개인화(personalization)를 뱐을 수 있는데, 이는 사이트에 접속하는 사용자 마다 원하는 형태의 모습과 구조로 변경이 가능하게 하여 보다 사용자의 취향에 맞는 정보를 제공하기 위한 노력이다. 이러한 포탈 개념이 기업 환경에 적용되면 기업 내의 다양한 정보들과 응용들을 통합하여 일관된 형식의 사용자 인터페이스를 제공하며, 개인화를 통해 자신이 원하는 업무 환경을 설정할 수 있는 응용을 생각해 볼 수 있다. 이러한 배경에서 EIP(Enterprise Information Portal)와 같은 응용이 제기 되었으며, 기존의 지식 관리 시스템의 효과적인 사용자 인터페이스를 위해 포탈 서버와 연계된 구조가 필요하게 되었다.

따라서 본 논문에서는 XML(Extensible Markup Language)을 이용하여 다양한 응용 프로그램들에 대한 표준화된 인터페이스를 제공하고, 이러한 응용들을 개인화시켜 제공할 수 있는 포탈 서버를 설계 및 구현하였다.

2. 관련 연구

EIP[1]는 기업 내의 각종 시스템과 정보를 통합하여 사용자에게 단일 접근점을 제공하며, 의사결정에 필요한 개인화된 정보를 제공할 수 있게 해주는 응용 프로그램이다. 포탈 서버는 EIP의 프리젠테이션 계층으로 사용자 인터페이스 통합 및 응용 통합을 수행한다. IBM, Oracle 등의 많은 회사들이 관련 제품들을 출시하고 있다[3],[4].

[2]에서는 이중의 시스템을 표준화된 웹 프로토콜을 이용하여 접근할 수 있게 하기 위해서 XML을 이용하여 응용 통합을 위한 인터페이스 정의 언어인 WIDL(Web Interface Definition Language)을 제안하였고, 이중의 데이터베이스나 레거시(legacy) 시스템 통합을 위한 XML 중계자에 대한 연구가 많이 진행되었다.

3. 포탈 서버 시스템 특징

3.1 응용 통합(Application Integration)

포탈 서버는 기업 내의 다양한 응용들을 일관된 형태의 사용자 인터페이스로 사용자에게 전달한다. 포탈 서버의 사용자는 개인 업무인 이메일 확인이나 일정관리뿐만 아니라 공동 작업인 보고서 작업 같은 다양한 종류의 응용들을 마치 하나의 통합 응용 프로그램과 같이 사용할 수 있게 되며, 모든 응용 프로그램들에 대해서 일관된 사용

자 인터페이스로 작업을 할 수 있다.

3.2 개인화(Personalization)

포탈 서버의 중요 요소로 지식 관리 시스템에서 제공하는 다양한 응용들을 자신이 원하는 방식으로 재배치 할 수 있고, 원하는 스타일로 변경을 할 수 있는 기능을 의미한다. 예를 들어 사용자는 자주 사용하는 업무 응용들을 한곳에 배치함으로써 업무 효율을 증대시킬 수 있으며, 자신이 원하는 스타일로 작업 환경을 변경할 수 있다.

3.3 웹 기반 사용자 인터페이스(Web-based User Interface)

사용자에게 익숙한 웹 기반의 사용자 인터페이스를 통해 보다 직관적이고 사용하기 편리한 업무 환경을 제공한다.

3.4 단일 로그인(Single Log-in)

기업 내에 다양한 응용들이 가지는 사용자 인증 정책을 통합 관리함으로써 사용자는 처음 한번의 인증 단계를 거침으로써 더 이상의 인증 과정 없이 기업 내의 모든 자원들에 대해 미리 주어진 권한에 따라 이용함으로써 사용자에게 편리함을 제공한다.

3.5 통합 검색(Unified Search)

기업 내에 분산되어 있는 정보 자원을 효율적으로 이용 가능하도록 각 정보에 대한 검색 기능을 통합 포탈 서버를 통해 일관된 검색을 할 수 있게 해준다.

4. 포탈 서버의 설계 및 구현

포탈 서버의 구성 요소는 그림 1과 같다. 사용자에게 보여지는 사용자 인터페이스 측면으로는 플레이스, 페이지, 포틀릿 등이 있으며, 이는 클라이언트 웹 브라우저에서 실행되어진다. 서버 측에는 이들을 지원하기 위한 3P(Place/Page/Portlet) 관리자와 사용자 관리 및 권한 제어를 하는 사용자 관리자가 있고, 포탈 서버에서 사용되어지는 메타데이터들을 저장하는 저장소와의 인터페이스 역할을 수행하는 메타데이터 관리자로 구성되어진다.

4.1 사용자 인터페이스

포탈 서버의 사용자 인터페이스는 개인화를 위한 핵심으로 포틀릿, 페이지, 플레이스 세가지 요소를 통해 이루어진다. 사용자는 원하는 모든 정보를 자신이 원하는 스타일로 구성하여 최적의 작업 환경을 구축할 수 있다.

4.1.1 포틀릿(Portlet)

포탈 서버에서 모든 응용 프로그램들은 포틀릿을 통해

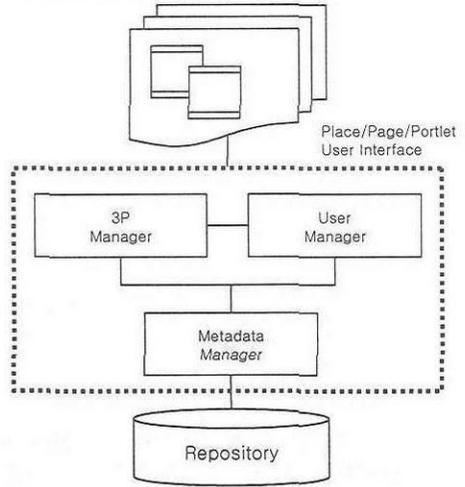


그림 1 포탈 서버 구조도

사용자에게 제공되어지므로, 사용자는 어떠한 정보나 응용 프로그램이라도 동일한 사용자 인터페이스를 통해 접근하게 된다. 즉, 포틀릿은 모든 응용 프로그램들에 대해 공통적인 인터페이스를 제공하는 “응용 윈도우(application window)”라고 할 수 있으며, 이동 및 크기 조절을 통해 자유로운 배치가 가능해서 사용자에게 개인화 서비스를 제공하게 된다. 예를 들어, 공동 프로젝트를 수행하는 경우 팀 전체의 수행 작업과 자신에게 국한된 수행 작업이 있을 수 있다. 이러한 경우 팀 작업 포틀릿과 개인 작업 포틀릿을 같이 배치하여 서로 비교해 볼 수 있으므로 업무 효율을 높일 수 있다.

4.1.2 플레이스, 페이지(Place/Page)

포틀릿이 생성되어지는 공간을 페이지라 하며, 서로 유사한 용도의 페이지 그룹들의 공간을 플레이스라고 한다. 사용자는 편의에 따라 새로운 페이지를 생성하고 해당 페이지에 포틀릿들을 배치할 수 있다. 플레이스는 개념적으로 분류 가능한 공간으로 정적인 공간과 동적인 공간이 있을 수 있다. 정적인 공간은 기업의 조직도처럼 미리 정해지는 그룹에 의한 공간이라 할 수 있고 동적인 공간은 프로젝트나 동호회처럼 소속이 계속 변하는 공간이라 할 수 있다.

4.2 3P(포틀릿, 페이지, 플레이스) 관리자

포틀릿, 플레이스 및 페이지의 생성, 삭제 등의 핵심 기능을 관리하며, 관리 대상으로는 포틀릿 패키지, 포틀릿 클래스, 플레이스, 페이지, 포틀릿 등이 있다. 3P 관리자의 구성요소로는 그림 2와 같이 포틀릿 패키지 관리자, 명세서 관리자, 포틀릿 생성 관리자 등이 있다.

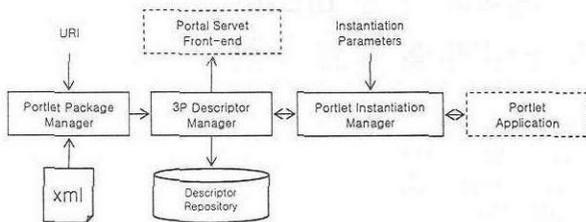


그림 2 3P 관리자

4.3 응용 통합

기존의 웹 기반의 기업 응용들을 포탈서버는 단일한 인터페이스를 통해 서비스할 수 있도록 한다. 이를 위해 포탈서버는 기존 기업 응용들을 통합할 수 있는 프레임워크를 구축하였다. 이 프레임워크는 크게 응용 등록, 포틀릿의 생명주기 관리, 보안/인증 관리로 구성된다.

4.3.1 응용 등록

포탈 서버는 포탈 서버 내의 각종 리소스를 그것에 대한 명세서(Descriptor)로 관리한다. 예를 들어 포틀릿 명세서는 포틀릿에 관한 모든 정보들을 담고 있는데, 이에 는 포틀릿 아이디, 이름, 설명, 응용 프로그램 정보, 위치 정보, 상태 정보 등이 포함된다. 서버에서는 이 명세서를 데이터베이스나 파일시스템에 영구적으로 저장하며, 클라이언트에서는 이 명세서를 서버의 EJB 컴포넌트인 세션 빈(Session Bean)을 통해 이용하게 된다.

포틀릿 패키지는 한가지 응용 프로그램을 위한 관련 포틀릿들의 집합이며, 하나 이상의 포틀릿을 가지게 된다.

명세서 관리자는 포탈서버가 동작하기 위한 여러 메타 정보를 관리하는 작업을 수행하는데 기본적으로 프레임워크, 페이지, 포틀릿에 대한 메타 정보를 관리하며, 이외에 포틀릿 패키지, 포틀릿 클래스에 대한 명세서도 관리 대상에 포함된다. 이 명세서 관리자는 포틀릿의 생명주기 관리, 보안/인증 모델을 위한 메타 정보를 제공해 준다.

4.3.2 포틀릿의 생명주기 관리

포탈서버는 포틀릿의 생명주기를 관리한다. 포틀릿은 필요에 따라 생성, 소멸되며, 이에 대한 관리를 포틀릿 생성 관리자와 포틀릿 삭제 관리자가 수행한다. 이 관리자는 필요에 따라 포틀릿 응용을 생성 혹은 삭제하기 위해 포틀릿 응용 생성자 또는 삭제자를 호출한다. 포틀릿 생명주기와 관련된 모든 필요 정보는 포틀릿 명세서 관리자로 부터 얻게 된다. 이러한 생명주기 관리자를 통해 사용자 혹은 포탈서버는 자유롭게 포틀릿을 통해 응용을 생성하여 사용하고, 삭제할 수 있는 유연성을 확보할 수 있게 된다.

4.3.3 보안/인증 모델

사용자 관리자는 사용자의 생성 및 삭제, 사용자의 권한 관리 및 로그인, 세션 관리, 개인화 정보 관리등을 수행한다. 이 사용자 관리자는 포탈서버와 응용간의 사용자 정보의 일관성을 유지하는 작업을 수행한다.

3P 관리자는 포탈서버의 여러 리소스에 대한 권한 및 보안 정보를 관리함으로써, 포탈서버의 보안을 유지하는 작업을 수행한다

4.4 메타데이터 관리자

메타데이터 관리자는 포탈 서버에서 사용되는 명세서, 사용자 정보, 개인화 정보 등의 자료들이 저장되는 저장소와의 인터페이스 역할을 하며 EJB의 엔티티 빈(Entity Bean)으로 구현되어 있다.

5. 구축예

개발한 포탈 서버 시스템은 클라이언트 부분과 서버 부분으로 나눌 수 있다. 사용자 인터페이스를 위한 클라이언트 부분은 DHTML 및 자바 스크립트를 이용하여 구현되어 있고, 서버 부분은 세션 관리 및 데이터베이스 인터페이스를 위한 EJB 서버 WebLogic의 컴포넌트들과 페이지 구성을 위한 JSP, 자바 서블릿들로 구성되어 있다. 각종 명세서들에 대한 저장소의 경우 권한 제어와 같이 빈 변환 요구가 필요한 것은 LDAP을 이용하였고, 그 외에는 Oracle 데이터베이스를 이용하였다.

6. 결론

포탈 서버는 기업 내의 다양한 시스템들에 대한 통합 및 개인화된 사용자 인터페이스를 통해 업무 효율을 극대화 시켜준다. 본 연구에서는 XML 명세서를 통한 응용 통합 및 포틀릿을 이용해 개인화된 사용자 인터페이스를 제공하는 포탈서버 시스템을 설계하고 구현하였다.

포탈 서버에서는 성능 상의 문제점은 개인화 서비스 제공으로 인해 발생하는 서버의 부담이다. 대부분의 페이지가 동적으로 생성되므로 브라우저의 캐쉬나 일반적인 서버 측의 캐쉬 방식에 의해서는 해결될 수 없다. 따라서 성능 향상을 위해 개인화된 페이지에 대한 캐쉬 방법에 대한 연구가 요구된다.

참고문헌

- [1] Christopher Shilakes, Julie Tylman, " Enterprise Information Portals" , Nov., 1998, Merrill Lynch
- [2] Charles Axel Allen, " WIDL, Application Integration with XML" , World Wide Web Journal 2(4): 229-248, 1997
- [3] WebSphere Portal Server, <http://www.ibm.com/webshere/portal>
- [4] Oracle Portal Server, <http://www.oracle.com/ip/dep/ias/portal/>