

IPlace 시스템에서의 공개 작업공간 지원

박희중[○] 김진홍 신원준 박양수 이명준

울산대학교 컴퓨터·정보통신공학부

{phj34[○], avenue, mathpf, yspark, mjlee}@ulsan.ac.kr

Supporting Open WorkSpaces In IPlace

Heejong Park[○] Jinhong Kim, Wonjoon Shin, Yangsu Park, Myungjoon Lee
School of Computer Engineering & Information Technology, University of Ulsan

요 약

IPlace 시스템은 등록된 사용자 사이에서 효과적인 공동작업을 지원하는 웹 기반 협업지원 시스템이다. 현재 개발된 IPlace 시스템은 자주 일어나는 공동작업을 등록된 사용자가 생성한 팀작업장을 통해서만 지원하고 있으므로 시스템에 등록하지 않은 사용자와의 효율적인 자료공유는 제공하지 못하고 있다.

본 논문에서는 이러한 문제점을 해결하기 위해서 IPlace 시스템에 등록하지 않은 사용자에게 효과적인 공동작업을 지원하기 위한 공개 작업공간인 OpenSpace에 대하여 기술한다. IPlace 시스템에 제한된 OpenSpace를 제공함으로써 기존의 IPlace 사용자와 등록되지 않은 사용자 사이의 자유롭고 편리한 자료공유 기능을 제공할 수 있다. OpenSpace는 플랫폼 독립적인 JSP 및 EJB 컴포넌트 기술을 이용하여 개발되었다.

장에서 결론 및 향후 연구과제에 대하여 기술한다.

1. 서 론

IPlace 시스템은 공동작업을 필요로 하는 그룹간의 공유 작업공간을 지원함으로써 여러 작업그룹내의 구성원이 이러한 환경에서 효과적으로 서로의 정보를 교환하고 공유할 수 있도록 지원해 주는 웹 기반 협업지원 시스템이다[1,2]. IPlace는 개인의 자료 공유를 위한 개인작업장, 자신이 속한 팀의 자료 관리를 위한 팀작업장 등을 지원한다. 특히 팀작업에 대해서는 사용자들은 웹이라는 가상공간에서 지역적인 제약 없이 정보의 공유를 통해 편리하게 공동작업을 할 수 있다는 장점이 있다[3]. 그러나 이런 작업들은 IPlace 사용자들 간에 로그인(log-in)을 통해서 사용할 수 있다는 제약이 있다. 이에 IPlace 사용자들은 다수의 비사용자들을 상대로 한 작업공간이 필요하게 되었다. 이를 위해 로그인없이 좀더 공개된 개념의 공동 작업공간인 OpenSpace(Open WorkSpace)를 설계 및 구현하였다.

OpenSpace는 다수의 IPlace 비사용자들을 대상으로 공개된 자료의 공유 및 교환기능을 제공하며, 또한 공개 작업공간의 성격을 설정해 줌으로써 자료의 공유와 관리를 용이하게 할 수 있도록 지원한다.

본 논문의 기반이 되는 IPlace 시스템은 플랫폼에 독립적인 JSP/EJB 기술[4]을 기반으로 개발되었다. 이에 OpenSpace도 JSP/EJB 기술을 바탕으로 개발되었으며, 공동작업에 참여하고 싶은 사용자는 누구나 파일의 업로드 및 다운로드, 주석달기, URL추가, 메모추가 등의 다양한 기능을 이용할 수 있게 된다.

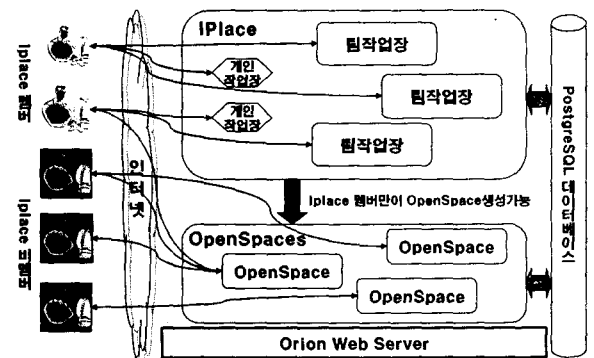
본 논문의 구성은 다음과 같다. 1장 서론에서는 IPlace 시스템과 OpenSpace의 개발배경에 대하여 설명하고 2장에서는 OpenSpace의 전체적인 구조와 특성에 대해서 살펴본다. 3장은 주요 컴포넌트의 설계 및 기능에 대해서 기술하며, 4장에서는 설계 내용에 대한 실제 구현된 기능을 중심으로 클라이언트측 구현 인터페이스를 보여준다. 5장은 OpenSpace의 활용 시나리오에 대해서 설명하고 마지막 6

2. OpenSpace 구조 및 특성

OpenSpace는 웹 기반의 공동작업 공간으로서 효율적인 자료의 공유와 유동적인 자료관리를 할 수 있는 환경이다. 또한 공동작업을 원하는 사용자들 사이에 파일의 업로드 및 다운로드, 주석달기, URL추가, 메모추가 등의 작업에 필요한 여러 가지 기능을 지원한다.

OpenSpace는 공동작업을 위한 공간 관리를 위해 플러더의 개념을 이용하여 자료를 효과적이고 체계적으로 분류할 수 있도록 제공한다. 본 장에서는 OpenSpace의 전체적인 구조와 특성에 대하여 기술한다.

2.1 OpenSpace의 구조



[그림 1] OpenSpace의 전체적인 구조

본 논문에서 개발된 OpenSpace는 공통의 목적이나 관심을 가진 다수의 사용자들이 웹상의 가상공간에서 정

보를 공유하고 교환할 수 있는 다양한 방법을 제공한다. OpenSpace는 다중사용자 환경을 지원하는 서버 클라이언트 구조로 다수의 사용자에 의한 자료의 이동 및 변동이 일어남으로 안정적으로 설계될 필요가 있다. 따라서 본 논문에서는 EJB 컴포넌트와 JSP를 이용하여 서버 시스템을 구현하였다. [그림 1]은 OpenSpace의 전체적인 구조를 보여준다. 그림에서 보는 바와 같이 먼저 IPlace 멤버가 IPlace를 통하여 특정한 OpenSpace를 만들면 IPlace의 멤버가 아닌 일반사용자들도 인터넷을 통해 이미 만들어진 OpenSpace에 참여하여 편리하게 공동작업에 필요한 자료를 공유할 수 있다.

2.2 OpenSpace의 특성

OpenSpace는 작업장의 성격에 따라 세 가지 타입의 작업장 생성기능을 제공한다. 작업장을 접근하는 사용자들은 누구나가 자료를 업로드하고 다운로드할 수 있는 "업로드 및 다운로드 지원 작업장", OpenSpace를 생성한 관리자가 웹을 통한 자료의 배포를 목적으로 하는 경우를 위한 "다운로드 only 작업장" 그리고, 자료의 업로드만을 허용하여 과제물의 수집이나 제출을 목적으로 하는 "업로드 only 작업장"으로 분류된다. 이렇게 다양한 타입의 작업장을 지원함으로써 효율적인 작업장의 운영 및 자료의 공유를 실현할 수 있게 된다.

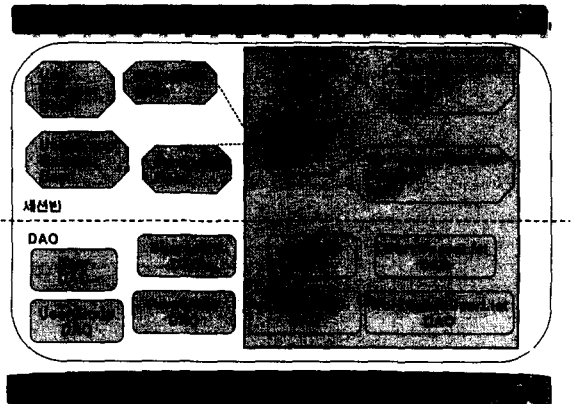
공동작업을 함에 있어서 방대한 양의 자료를 효과적이고 편리한 관리를 위해 OpenSpace는 폴더의 개념을 이용하고 있다. 폴더를 이용하면 많은 자료들의 분류를 편리하게 할 수 있다. 그리고 관리자가 복사하기, 잘라내기, 붙여넣기 등의 기능을 이용하여 자료의 이동을 쉽게 함으로써 자료를 유동적으로 관리를 할 수도 있다. 무분별한 폴더의 생성을 지양하기 위하여 폴더의 생성은 작업장 생성자에 의해서만 관리되도록 구성하였다.

3. OpenSpace의 설계

본 장에서는 OpenSpace를 설계함에 있어서 중요한 역할을 하는 두 가지 즉, 사용자와 관리자의 요청 액션에 따라 해당 작업을 수행하는 EJB 컴포넌트와, 자료의 영속성 유지를 위한 데이터베이스의 구성에 대하여 기술한다.

3.1 EJB 컴포넌트의 설계

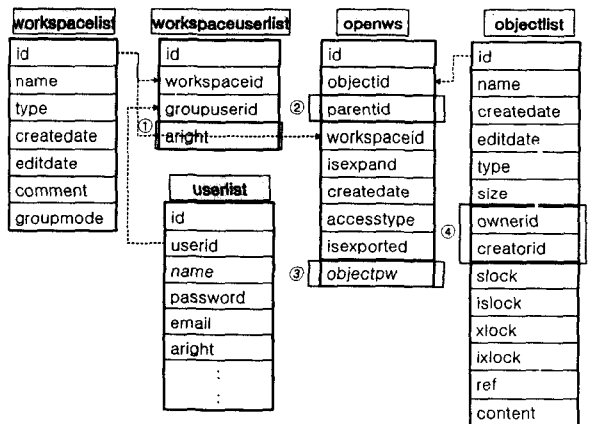
EJB 컴포넌트는 [그림 2]와 같이 사용자나 관리자의 요청 액션이 발생했을 때 그 액션에 해당하는 세션빈이 호출된다. 그 세션빈은 직접 데이터베이스에 접속하지 않고 자신과 같은 이름을 가진 DAO(Data Access Object) 객체를 이용하여 데이터베이스에 내용을 삽입, 수정, 삭제 등의 일을 수행한다. 즉 세션빈은 사용자의 액션에 대한 처리를 담당하며 DAO객체는 데이터베이스와의 연결코드와 쿼리 생성, 실제 데이터베이스의 변경 등의 코드를 캡슐화 하고 있다.



[그림 2] OpenSpace의 협업지원 컴포넌트 구조

3.2 데이터베이스의 설계

OpenSpace를 위한 데이터베이스 테이블은 [그림 3]과 같이 모두 5개로 이루어져 있다. userlist는 IPlace 사용자에 대한 고유 id와 사용자 정보를 저장한다. workspaceuserlist는 생성되어 있는 OpenSpace의 목록을 가지고 있으며, workspaceuserlist에는 OpenSpace를 만든 관리자의 정보를 가지고 있다. 그리고 objectlist는 실제 저장되어 있는 자료에 대한 정보, openws에는 objectlist에 저장된 자료가 어느 OpenSpace의 정보인지와 그 자료에 대한 정보가 저장된다.



[그림 3] 데이터베이스 설계

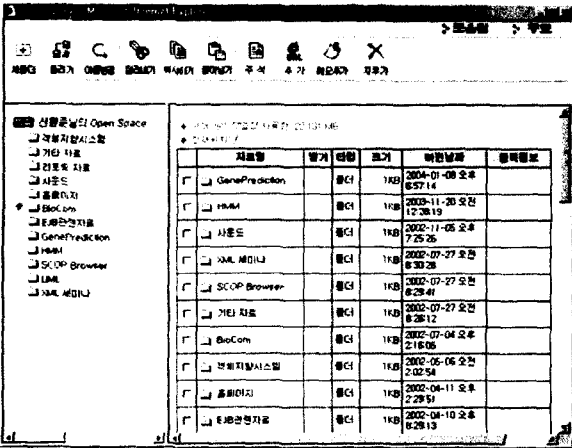
각각의 데이터베이스 테이블에서 ①은 OpenSpace의 관리자인지 일반 사용자인지를 파악하기 위한 필드이며 ②는 objectlist에 저장된 자료가 파일인지 URL인지 폴더인지에 대한 구분을 위한 정보를 저장한다. ③은 저장된 자료의 수정과 삭제를 위한 password가 저장되며 ④는 자료를 올린 사용자나 그 자료에 대한 소유권이 있는 사용자에 대한 id가 저장된다.

4. OpenSpace의 구현

OpenSpace는 사용자에게 따라 수행할 수 있는 기능이 분리되어 있다. 즉 OpenSpace를 만든 관리자는 폴더의 생성과 이동, 삭제 등의 기능이 가능하고 일반 사용자는 메모나 주석, URL은 마음대로 사용이 가능하지만 파일의 업로드와 다운로드에 대해서는 OpenSpace를 만들 때 설정한 속성에 따라 사용할 수 있는 권한이 다르다.

[그림 4]에서 보는 것은 OpenSpace의 관리자 인터페이스를 나타낸다. 사용자의 편의를 위해 메뉴를 사용하지 않고 아이콘을 이용해 관리를 위한 기능을 쉽게 사용할 수 있도록 구현하였다. 그리고 왼쪽 프레임에는 폴더를 오른쪽 프레임에는 폴더와 현재 업로드 되어 있는 자료의 리스트를 보여주도록 하였다.

사용자 중심의 인터페이스 설계에 초점을 두었으며 윈도우 탐색기와 유사한 인터페이스를 구현함으로써 처음 이용하는 사용자도 용이하게 제공되는 서비스를 이용할 수 있도록 하였다.

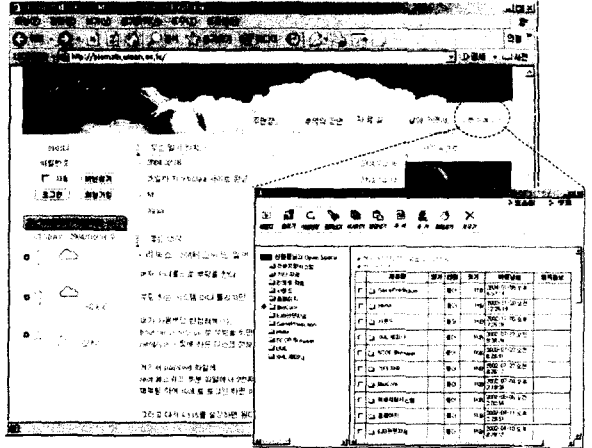


[그림 4] OpenSpace 인터페이스

5. OpenSpace의 활용

OpenSpace는 개인작업에 대한 자료관리 뿐만 아니라 웹상의 여러 사용자의 도움이 필요한 작업을 할 때도 유용하게 사용될 수 있다. 그리고 방대한 자료에 대해서도 효율적으로 분류를 할 수 있고 이동이나 삭제 등이 편리하므로 효과적인 자료공유나 유동적인 자료관리를 위해 적절하게 활용되어 질 수 있다. 예를 들어 [그림 5]와 같이 개인 홈페이지에 OpenSpace를 포함하여 자신의 홈페이지에 접속하는 웹 사용자들에게 자료를 제공하거나 자신이 자료를 제공받을 수도 있다. 또한 OpenSpace를 생성할 때 설정할 수 있는 세 가지 속성을 이용하면 사용자가 파일을 올릴 수만 있게 한다면, 파일을 다운로드만 할 수 있게 하는 기능을 적절히 이용하여 개인적인 자료관리나 중요한 문서에 대한 보호에도

활용되어 질 수 있다.



[그림 5] 개인 홈페이지에 포함된 OpenSpace

6. 결론

본 논문에서는 웹상에서 누구나 이용할 수 있고 효율적인 자료공유와 유동적인 자료관리를 위한 공간인 OpenSpace를 설계하고 구현하였다. 구현된 OpenSpace는 공동작업을 필요로 하는 사용자들 간의 자료와 정보를 공유하는 방법을 보다 원활하고 효과적으로 수행되어 질 수 있도록 도와준다. OpenSpace는 플랫폼 독립적인 JSP/EJB 기술을 이용하여 구현되었으며 세 가지 다른 속성을 지닌 OpenSpace를 지원해 줌으로써 시스템의 활용성을 높였다. 폴더의 개념을 이용하여 자료의 편리한 분류를 가능하게 하였고 단순히 자신이 만든 OpenSpace의 링크정보를 이용하여 개인 홈페이지나 다른 웹 페이지에도 쉽게 포함 할 수 있도록 구현하였다. 향후 연구과제로는 OpenSpace 전체에 대한 속성 설정 뿐 만 아니라 다양하게 생성되는 폴더의 속성을 설정할 수 있는 기능을 추가하여 좀더 세분화된 폴더 관리에 대한 활용도를 높일 예정이다.

7. 참고문헌

- [1] Gyu-Wan Kim, Soo-Kwon Jeong, Jae-Hoon Jeong, In-Ho Kim, Myung-Joon Lee, "CoDocs : An electronic document management system supporting effective collaborative work," 8th International conference On Human-Computer Interaction, 1999
- [2] 안건태, 정명희, 이근웅, 문남두, 이명준, "iPlace:EJB 기술을 이용한 웹 기반 협업시스템", 정보처리학회논문지 D 제 8-D권 제 6호(2001년 12월)
- [3] 정명희, 안건태, 문남두, 김진홍, 이명준 "웹 기반 공동작업공간의 지원을 위한 EJB 컴포넌트의 개발" 한국정보과학회 '2001 봄 학술발표논문집(B) 제 28권 1호, p.469-471, 2001
- [4] EJB™ Tech "Server Component Model for the Java™ Platform" by Anne Thomas, December 1998, Prepared for Sun Microsystems, Inc.