

개인화된 서비스를 지원하는 웹데브 기반의 협업시스템

이 흥 창*, 이 명 준**

A WebDAV-based Collaborative System Supporting Individualized Services

Hong-Chang Lee *, Myung-Joon Lee **

요 약

인터넷을 통하여 다양한 활동을 수행하는 사용자들이 급속히 증가함에 따라 여러 그룹에 소속된 사용자들 간의 인터넷을 통한 협업과 정보 교환이 활발하게 이루어지고 있다. 또한, 이와 더불어 최근 사용자 중심의 서비스가 활성화됨에 따라 개인화된 인터넷 서비스의 중요성이 대두되고 있다. 현재, 웹 기반의 협업 수행을 지원하는 다양한 협업시스템이 있다. 하지만 이러한 협업시스템들은 개인화된 서비스를 지원하지 않기 때문에 다양한 사용자를 고려한 효과적인 협업 환경을 제공하기 어렵다.

본 논문은 CoSlide 협업시스템을 이용하여 다양한 사용자들의 협업 수행을 지원하기 위한 협업지원 포틀릿의 개발에 대하여 기술한다. 협업지원 포틀릿은 CoSlide 협업시스템 서버에서 제공하는 다양한 가상공간을 지원하며, 각 가상공간의 자원을 제어할 수 있는 여러 기능들을 제공한다. 개발된 포틀릿은 포털을 구성하는 컴포넌트로 포함되어 다양한 사용자의 요구를 충족시킬 수 있는 협업 환경을 제공한다. 사용자는 다양한 포틀릿을 제공하는 포털을 통하여 자신의 작업 환경을 고려한 맞춤형 인터페이스를 구성하고, 이를 이용하여 효과적으로 협업을 수행할 수 있다.

Abstract

As internet users swiftly increased in number, both collaborations and communications on the Internet gradually increased. Accordingly, the importance of individualized service has been raised as user-centered service has been activated. As of now, there are many systems supporting collaboration based on the Web. Unfortunately, since the existing Web-based collaborative systems do not support individualized service, they do not provide effective collaboration for various kinds of users.

In this paper, we describe the development of a collaboration portlet for the CoSlide collaborative system. The collaboration portlet provides various methods which control the resources of each workspace on the CoSlide server. As a component of a portal, the developed portlet provides effective collaborative environment satisfying various user requirements. Users can collaborate through the individualized interfaces which are composed by portlets on a portal for their own working environment.

• 제1저자 : 이흥창 교신저자 : 이명준

• 접수일 : 2008. 4. 29, 심사일 : 2008. 5. 29, 심사완료일 : 2008. 9. 25.

* 울산대학교 컴퓨터정보통신공학부 박사과정 ** 울산대학교 컴퓨터정보통신공학부 교수

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 육성지원사업의 연구결과로 수행되었습니다. (IITA-2008-(C1090-0801-0039))

▶ Keyword : 협업지원 포틀릿(Collaboration Portlet), CoSlide 협업시스템(CoSlide Collaborative System), 포틀릿(Portlet), 포탈(Portal)

I. 서 론

인터넷을 통하여 다양한 활동을 수행하는 사용자들이 급속히 증가함에 따라 여러 그룹에 소속된 사용자들 간의 인터넷을 통한 협업과 정보 교환이 점차 늘어나고 있다. 또한, 이와 더불어 최근 사용자 중심의 서비스가 활성화됨에 따라 개인화된 인터넷 서비스의 중요성이 대두되고 있지만 기존의 협업시스템들은 사용자를 고려한 사용자 중심의 서비스를 제공하지 않아서 효과적인 협업 환경을 제공하기 어렵다.

포탈(Portal)[1,2]은 여러 시스템의 정보를 포틀릿(Portlet)[3] 단위로 정의하여 웹 기반의 통합 서비스를 제공하는 게이트웨이이다. 포탈은 다양한 포틀릿을 통하여 사용자별로 인터페이스와 어플리케이션을 지원함으로써 개인화된 서비스를 제공하며, 그에 따라 점차 그 사용 기반을 넓혀가고 있다. 포틀릿은 포탈을 구성하는 컴포넌트로 개별적으로 정보를 처리하고 화면을 구성하여 포탈의 다양성을 지원해주는 핵심 기술이다. 포탈은 포틀릿을 통하여 여러 시스템의 정보를 통합하고 다양한 인터페이스를 구성하여 사용자별로 제공한다.

Jakarta Slide[4]는 아파치 웹 서버에서 동작하는 모듈로서 WebDAV[5,6,7] 프로토콜을 통하여 다양한 콘텐츠의 비동기적인 협업 환경을 제공한다. 사용자는 Jakarta Slide 기반의 협업시스템을 통하여 여러 사용자와 원거리에 위치한 자원을 공유하면서 비동기적인 협업을 수행할 수 있다. CoSlide[8,9]는 Jakarta Slide를 기반으로 개발된 협업시스템으로서 사용자와 그룹을 체계적으로 지원하는 가상공간을 제공하여 보다 효과적인 협업 수행을 지원한다. 기존의 Jakarta Slide는 개별 사용자와 사용자 그룹에 대한 가상공간을 체계적으로 지원하지 않아 다양한 그룹을 통하여 다수의 사용자가 협업을 수행하는 환경을 제공하는 것은 매우 어려운 작업이었다. CoSlide는 Jakarta Slide의 기능을 바탕으로 사용자와 그룹을 위한 별도의 가상공간을 제공하고 가상공간별로 접근권한을 부여하여 다양한 사용자와 그룹이 참여하는 협업을 위한 환경을 체계적으로 제공한다. 사용자는 CoSlide 협업시스템에서 제공하는 협업 정보를 활용하기 위하여 CoDAView[10,11]와 같은 협업시스템 전용 클라이언트를 사용할 수 있다. 하지만 전용 클라이언트는 플랫폼에 종속적이며 다양한 사용자의 작업 환경을 고려한 인터페이스 및 기능을 제공하기 어렵다. 따라서 다양한 사용자의 작업 환경에

능동적으로 포함되어 편리하게 CoSlide 협업시스템을 사용할 수 있는 웹 기반의 협업 도구가 필요하다.

본 논문에서는 CoSlide 협업시스템 서버를 이용하여 다양한 사용자들의 협업 수행을 지원하기 위한 협업지원 포틀릿의 개발에 대하여 기술한다. 협업지원 포틀릿을 효과적으로 개발하기 위하여 MVC 모델 기반의 구조적인 웹 어플리케이션 모델링을 지원하는 Struts[12] 프레임워크를 바탕으로 포틀릿 구조를 설계하였으며, 포탈에서 서비스될 수 있도록 JSR-168 표준에 준하여 각 모듈을 구현하였다. 협업지원 포틀릿은 CoSlide 협업시스템 서버에서 제공하는 다양한 가상공간을 지원하며, 각 가상공간의 자원을 제어할 수 있는 여러 기능들을 제공한다. 개발된 포틀릿은 포탈을 구성하는 컴포넌트로 포함되어 다양한 사용자의 요구를 충족시킬 수 있는 협업 환경을 제공한다. 포탈은 포틀릿들로 구성된 다양한 인터페이스를 제공하며, 사용자는 자신의 작업 환경을 고려한 맞춤형 인터페이스를 통하여 효과적으로 협업을 수행할 수 있다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 관련 연구에 대하여 살펴보고 3장에서는 협업지원 포틀릿의 설계와 설계를 바탕으로 한 구현을 살펴본다. 4장에서는 개발된 포틀릿과 CoSlide 협업시스템에 대하여 평가하고 마지막으로 5장에서 결론을 살펴보고자 한다.

II. 관련 연구

2.1 포탈(Portal)

포탈은 개인이나 그룹이 여러 시스템으로 나뉘서 관리하는 정보들을 통합하여 개인이나 그룹의 활동에 필요한 모든 자원을 하나의 웹 화면으로 제공하고 관리, 분석할 수 있는 시스템이다. 포탈은 웹 기반의 게이트웨이로서 다양한 사용자와 그룹의 요구에 상응하는 맞춤형 인터페이스를 제공하고 그에 대응하는 어플리케이션을 지원하여 개인화된 맞춤 서비스를 제공한다. 포탈은 이러한 개인화된 서비스와 사용자 중심의 인터페이스를 제공하기 위하여 포틀릿이라는 컴포넌트를 사용한다. 포틀릿을 활용하여 사용자들을 고려한 다양한 서비스를 제공하는 포탈 서버로는 Sun의 Portal Server 7[13], IBM의 WebSphere[14], Oracle의 OracleAS[15] 등과 함께 오픈 소스 서버인 Liferay Portal[16]이 널리 이용되고 있다.

2.1.1 Liferay Portal

Liferay는 가장 널리 활용되고 있는 오픈 소스 엔터프라이즈 포탈로서 Java, J2EE, Web2.0 등의 기술을 기반으로 개발되어 사용자들을 고려한 다양한 포탈 기능을 제공하고 포틀릿 컨테이너로서 포틀릿을 활용하여 다양한 기능을 제공하기 위한 기반을 지원한다. 또한, 주요 웹 서버와 서블릿 컨테이너와 쉽게 융합되어 동작하며 다양한 데이터베이스와 운영체제를 지원하여 플랫폼 독립적으로 구축될 수 있다. <표 1>은 Liferay 포탈의 주요 특징을 나타낸다.

표 1. Liferay 포탈의 주요 특징
Table 1. Features of the Liferay Portal

- JSR-168 표준 포틀릿 스펙 지원
- 60가지가 넘는 다양한 포틀릿 제공
- 편리한 개발 도구 제공
- 콘텐츠 관리 시스템(CMS) 지원
- AJAX를 지원하는 사용자 인터페이스
- 약 22여 언어 지원
- 완벽한 LDAP 동기화 지원
- MIT 라이선스 제공
- 룩 앤 필(Look and Feel) 원클릭 적용
- 실시간 포틀릿 렌더링
- 향상된 테마/레이아웃(Layout) 기능

2.2 포틀릿

2.2.1 포틀릿

포틀릿은 웹 화면에서 동적인 컴포넌트 구성을 지원하는 환경으로 정적인 인터넷 환경에서 개인화된 서비스를 제공할 수 있는 핵심 기술이다. 초기 웹 환경에서는 정적인 인터넷을 기반으로 개인화된 서비스를 제공하기 위한 특별한 규약이 없어서 서비스 제공자별로 컴포넌트가 복잡하게 개발되어 제공되었으나, 2003년에 Sun을 주축으로 웹 기반의 포틀릿을 정의하고 효과적인 포틀릿을 개발하기 위하여 JSR-168이라는 Java 기반의 표준이 마련되었다.

포틀릿은 서블릿과 같은 웹 화면을 구성하는 컴포넌트이지만 서블릿이 하나의 페이지를 정의하는 단위임에 반해, 포틀릿은 하나의 페이지를 구성하는 여러 요소 중의 한 요소를 정의하는 마크업 단위이다. 서블릿으로 구성된 웹 화면은 사용자들에게 동일한 서비스를 제공하지만 포틀릿으로 구성된 웹 화면은 사용자의 요구별로 다양한 포틀릿을 조합하여 차별된 서비스를 제공할 수 있다. 또한, 포틀릿은 포탈이나 웹 서버 같은 웹 어플리케이션에 쉽게 결합되어 사용될 수 있는 고유의 속성들을 가지고 있다. 이러한 고유의 속성들은 포틀릿을 관리하는 포틀릿 컨테

이너에 의하여 관리되고 제어된다. <그림 1>은 하나의 웹 페이지를 구성하는 다양한 포틀릿의 조합을 보여준다.[17]

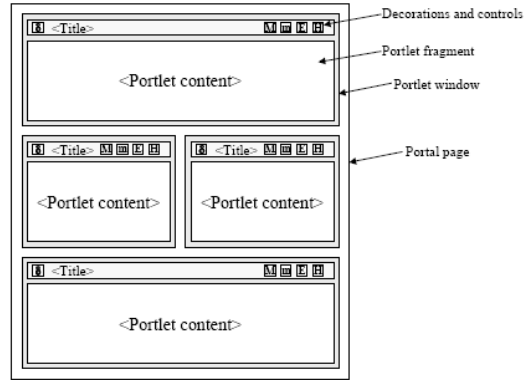


그림 1. 포탈 페이지를 구성하는 포틀릿
Fig 1. Portlets on a Portal Page

2.2.2 포틀릿 컨테이너

포틀릿 컨테이너는 포틀릿에 관한 클라이언트, 혹은 서버의 요청이 있을 때 대상 포틀릿과 그 포틀릿의 프로바이더를 실행하고, 요청하는 곳으로 포틀릿 서비스를 제공한다. 포틀릿 컨테이너는 포틀릿이 실행될 수 있는 기본 환경을 제공하며 포틀릿이 실행되고 종료될 때까지의 생명주기에 관여하며, 포틀릿 고유의 속성들을 제어하거나 관리한다.

포틀릿 컨테이너에서 포틀릿이 실행될 때, 포틀릿 컨테이너는 포틀릿을 위한 콘텐츠와 정보들을 전달받고 이를 요청한 곳으로 전달하는 중계 역할을 한다. 포탈 서버는 포틀릿 컨테이너로부터 전달받은 콘텐츠와 정보를 바탕으로 사용자의 요청에 부합하는 포탈 페이지를 생성하여 제공한다.

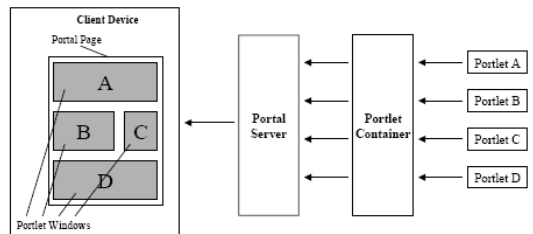


그림 2. 포탈 페이지 생성 과정
Fig 2. Portal Page Creation

<그림 2>는 포틀릿이 생성되어 제공되는 과정을 보여준다.[18] 포틀릿 컨테이너는 요청된 여러 포틀릿들을 실행하고 그 결과를 포탈 서버에게 전달한다. 포틀릿 컨테이너로부

터 전달받은 정보를 바탕으로 포탈 서버는 사용자의 요구사항에 맞도록 페이지를 구성하여 제공함으로써 사용자별로 개인화된 서비스를 제공할 수 있다.

2.3 Struts 프레임워크

Struts는 아파치 프로젝트의 일부로서 웹 어플리케이션의 구조적인 개발을 위하여 널리 사용되는 프레임워크이다. Struts는 웹 어플리케이션의 효과적인 개발 및 유지보수를 위하여 MVC(Model-View-Controller) 모델 기반의 구조적인 모델링 환경을 제공한다.

표 2. Struts의 MVC 모델 구성요소
Table 2. Components of the Struts Framework

Model	어플리케이션의 비즈니스 로직을 구현하는 부분, 처리해야 할 데이터와 데이터를 다루는 연산의 집합, 데이터베이스와 연결
View	어플리케이션의 프레젠테이션 로직을 구현하는 부분, 처리된 데이터를 사용자의 요구에 맞도록 화면에 보여주는 영역으로 태그라이브러리를 통하여 효과적으로 구현을 지원함
Controller	뷰 영역과 모델 영역 간의 정보 흐름을 정의하고 관리하는 역할, 사용자의 요구에 맞게 알맞은 뷰와 모델을 결정하고 정보를 전달

〈표 2〉는 Struts를 구성하는 각 영역의 설명을 나타낸다. Struts는 MVC, 각 영역의 독립적인 구현을 보장한다. 따라서 Struts 구조를 기반으로 개발된 웹 어플리케이션은 영역별로 독립적으로 구현되어 있어 추후 확장 및 유지보수가 용이하여 유연하게 관리될 수 있다.

2.4 CoSlide 협업시스템

2.4.1 WebDAV 프로토콜

WebDAV는 인터넷을 통하여 다양한 콘텐츠의 비동기적인 협업 작성을 지원하기 위한 프로토콜이다. WebDAV는 HTTP/1.1을 확장하여 사용자들에게 원격지에 위치한 자원을 제어하고 관리할 수 있는 다양한 기능을 제공한다. WebDAV는 HTTP/1.1에서 제공되는 기본 메서드를 지원하며 이와 더불어 새로운 메서드를 추가, 확장하여 속성을 이용한 자원 관리, 속성 관리, 잠금 관리, 덮어쓰기 방지, 이름 공간 관리 등의 자원 제어 기능을 제공한다.

2.4.2 CoSlide 협업시스템

CoSlide는 WebDAV 프로토콜을 사용하여 사용자와 그룹을 위한 가상공간을 지원하여 효과적으로 협업서비스를 제공하는 협업시스템이다. CoSlide 협업시스템은 개인 사용자를 위한 가상공간인 개인작업장, 다양한 사용자가 접근하여 공개된 작업을 수행할 수 있는 가상공간인 공개작업장, 그리고 특정 그룹에 소속된 사용자들 간의 효율적인 그룹 작업을 지원하는 그룹작업장을 제공한다.

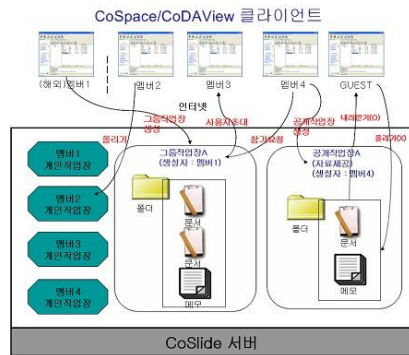


그림 3. CoSlide 협업시스템의 동작
Fig 3. Works on the CoSlide Collaborative System

〈그림 3〉은 CoSlide 협업시스템의 전체적인 동작을 보여준다. CoSlide 협업시스템은 Jakarta Slide를 확장하여 개발된 CoSlide 서버와 CoSlide 서버에서 제공되는 협업서비스를 활용하는 클라이언트로 구성되며 현재 윈도우즈 클라이언트(8)와 리눅스 클라이언트인 CoDAView가 개발되어 있다.

III. 협업 수행을 지원하는 포틀릿 개발

본 장에서는 효과적인 협업 환경을 제공하기 위하여 사용자 중심의 인터페이스를 지원하는 협업지원 포틀릿의 개발에 대하여 기술한다. 협업지원 포틀릿은 포탈에 포함되어 효과적으로 협업 정보들을 제공하여야 하며, WebDAV 프로토콜을 이용하여 CoSlide 협업시스템 서버에 접근하고 협업 정보를 처리하기 위한 요청을 처리해야 한다. 협업지원 포틀릿은 CoSlide 협업시스템 서버와 연동하여 다양한 가상공간과 이를 제어하는 기능을 제공하여 효과적으로 협업 수행을 지원한다. 이와 같은 다양한 기능을 지원하기 위한 여러 모듈을 효과적으로 구성하기 위하여 Struts 프레임워크를 사용한다. 협업지원 포틀릿은 Struts 프레임워크를 사용하여 프레젠테이션 로직과 비즈니스 로직으로 구성되며 각 영역의 유기적인

연결이 정의된다. <그림 4>는 개발된 협업지원 포틀릿을 통하여 다양한 사용자에게 사용자 중심의 인터페이스를 제공해주는 포탈의 서비스 제공 과정을 보여준다.

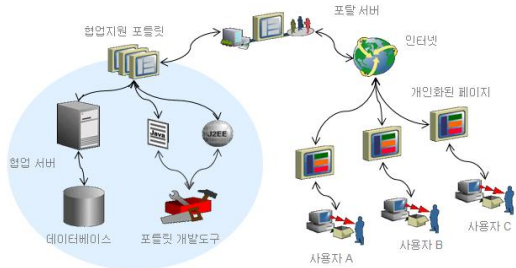


그림 4. 포탈을 이용한 협업서비스 제공 과정
Fig 4. Collaboration Services on Portal

3.1 협업지원 포틀릿의 설계

협업지원 포틀릿은 Struts 프레임워크의 MVC 모델링에 따라 모델, 뷰, 컨트롤러 영역을 구현한다. 모델 영역은 WebDAV API를 사용하여 협업 정보를 처리하는 비즈니스 로직을 정의하며, 뷰 영역은 Struts 태그라이브러리를 이용하여 협업 정보를 표현하는 프레젠테이션 로직을 정의한다. 또한, 컨트롤러 영역은 모델과 뷰 영역의 연결과 데이터 흐름을 정의한다.

3.1.1 협업지원 포틀릿의 구조

협업지원 포틀릿은 협업정보를 효과적으로 처리하고 그 결과를 제공하기 위하여 모델과 뷰 영역을 구현하고, 각 영역의 연결과 데이터 흐름을 유기적으로 정의하기 위하여 컨트롤러 영역을 구현한다.

(1) 협업지원 포틀릿의 모델 영역

협업지원 포틀릿의 모델은 CoSlide 협업시스템 서버와 연결하고 협업 정보를 처리하는 비즈니스 로직이 정의되어 있는 영역으로 뷰 영역에서 전달된 요청을 처리하는 역할을 하는데 주로 서버에 접근하고 가상공간의 자원을 제어하기 위한 WebDAV 메시지를 호출한다. CoSlide 협업시스템 서버로부터 전달받은 협업 정보는 컨트롤러 영역에 정의된 흐름에 따라 새로운 뷰 영역으로 전달된다.

(2) 협업지원 포틀릿의 뷰 영역

협업지원 포틀릿의 뷰는 협업 정보를 표현하는 프레젠테이션 로직이 정의되어있는 영역으로 사용자가 접근할 수 있는 가상공간의 정보를 나타내는 작업장 트리 부분과 선택한 작업장 하위의 자원 정보를 보여주고 자원 제어 기능을 제공하는 작업장 부분으로 구분된다. 작업장 트리 부분은 모델

영역에서 처리된 협업 정보를 전달받아 현재 사용자가 접근할 수 있는 가상공간인 개인작업장, 공개작업장 그리고 그룹작업장의 정보를 트리 형태로 보여준다. 사용자는 트리 형태의 익숙한 GUI로 표현되는 가상공간을 간편하게 확인하고 접근할 수 있다. 그리고 작업장 부분은 사용자가 작업장 트리에서 선택한 작업장의 자원 정보와 자원을 제어할 수 있는 기능들을 나타낸다. 사용자가 작업장 부분에서 자원의 처리를 요청하게 되면, 컨트롤러 영역에 정의된 흐름에 따라 적절한 모델 영역으로 그 요청이 전달된다.

(3) 협업지원 포틀릿의 컨트롤러 영역

협업지원 포틀릿의 컨트롤러는 모델과 뷰 영역의 연결과 데이터의 흐름을 정의하는 영역이다. Struts 프레임워크의 구조에 따라 컨트롤러 영역은 별도의 XML 문서를 통하여 정의되며, 이는 추후 서비스의 확장이나 영역 간의 새로운 데이터 흐름을 소스 코드와는 별개로 XML 문서로 정의하여 소스 코드를 수정하지 않고 손쉽게 적용할 수 있다.

3.2 협업지원 포틀릿의 구현

포탈이 제공하는 사용자 중심의 인터페이스에 포함되어 개인화된 서비스를 제공하고 효과적으로 CoSlide 협업시스템과 연동하기 위하여 Struts 구조로 설계된 협업지원 포틀릿의 각 영역별 세부 구현이 필요하다. <그림 5>는 포탈에 포함되어 사용자의 요청을 처리하는 협업지원 포틀릿의 내부 동작을 보여준다.

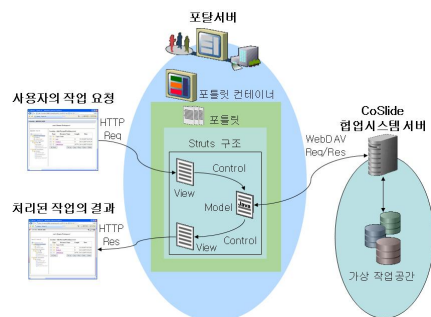


그림 5. 협업지원 포틀릿의 내부 동작
Fig 5. Operations on Collaboration Portlet

3.2.1 협업지원 포틀릿의 모델 영역 구현

협업지원 포틀릿에서 비즈니스 로직을 담당하는 모델 영역은 CoSlide 협업시스템 서버에 접근하여 협업 정보를 주고받는 서블릿 클래스로 구성된다.

(1) 서블릿 클래스의 구현

협업지원 포틀릿의 모델 영역을 구성하는 서블릿 클래스는 뷰 영역에서 전달받은 정보를 바탕으로 CoSlide 협업시스템 서버에 접근하고 해당 정보를 요청하고 그 결과를 처리한다

음, 뷰 영역으로 전달하는 역할을 한다.

① CoSlide 협업시스템 사용자 인증

```
// WebDAV API 포함
import edu.uci.ics.DAVExplorer.*;

public String getServerInfoFor( ... ){
    DefaultAuthHandler.setAuthorizationPrompeter(...);
    CookieModule.setCookiePolicyHandler( null );
    // 협업시스템 서버에 연결하는 클래스 생성
    ws = new WebDAVConnection(serverAdd,8080);
    // 사용자 정보를 통하여 인증
    ws.addBasicAuthorization( ... , name, pass);
    // 접속 및 요청에 필요한 정보
    NVPair[] Headers = new NVPair(3);
    Headers(0) = new NVPair("Host", serverAdd);
    Headers(1) = new NVPair("Depth", "infinity");
    ....
}
```

그림 6. CoSlide 협업시스템 서버에 접근
Fig 6. Connection to the CoSlide Collaborative Server

〈그림 6〉은 서블릿 클래스에서 WebDAV API를 사용하여 CoSlide 협업시스템 서버에 접속을 시도하는 코드를 보여준다. 이 코드는 뷰 영역에서 전달받은 사용자의 정보를 바탕으로 WebDAV API에서 제공하는 WebDAVConnection 클래스의 객체를 통하여 연결을 시도하고 사용자 인증을 요청하는 내용을 보여준다.

② 접근 가능한 작업장의 정보 요청

```
String s = <D:propfind xmlns:D="DAV:">
    + ... </D:propfind>;
byte[] c = s.getBytes();
// WebDAVConnection 클래스 객체의 메서드를 통한 서비스 요청
HTTPResponse h =
    ws.PropFind("/slide/OpenWorkspace", c, Headers);
// 요청 결과를 XML 형태로 전달 받음
String result =
    getData(new ByteArrayInputStream(h.getData()));;
```

그림 7. WebDAV를 이용한 협업 정보 요청
Fig 7. Request the Information using WebDAV

〈그림 7〉은 CoSlide 협업시스템 서버와의 연결 정보를 가지고 있는 WebDAVConnection 클래스 객체의 PropFind 메서드를 통하여 특정 작업장에 접근하여 협업 정보를 요청하는 코드를 보여준다.

③ XML 정보의 변환

```
// 전달받은 XML 형태의 정보를 XML 파서를 통하여 변환, 변환 후
// 작업장 정보를 분류
NodeList nodeResponse =
    doc.getElementsByTagName("D:response");
for(int i=0;i<nodeResponse.getLength();i++){
    NodeList nodeResponseChild =
        nodeResponse.item(i).getChildNodes();
    uridata = new URIData();
    // 협업 정보 중에서 작업장 목록을 추출
    for(int j=0;j<nodeResponseChild.getLength();j++){
        String name =
            nodeResponseChild.item(j).getNodeName();
        if(name.equals("D:href")){
            NodeList nodeHref =
                nodeResponseChild.item(j).getChildNodes();
            for(int k=0;k<nodeHref.getLength();k++){
                // 작업장 목록에서 작업장 이름 추출
                int type = nodeHref.item(k).getNodeType();
                if(type == Node.TEXT_NODE)
                    uridata.setURI(
                        nodeHref.item(k).getNodeValue());
            }
        }
        ...
    }
}
```

그림 8. XML 형태의 협업 정보 변환
Fig 8. Parsing the Information in XML Format

〈그림 8〉은 CoSlide 협업시스템 서버로부터 받은 XML 형태의 협업 정보를 뷰 영역에서 처리할 수 있는 클래스 형태로 변환하는 코드를 보여준다. XML 형태로 전달받은 데이터는 트리 형태로 정보를 표현하며, 이 정보는 DOM 파서를 이용하여 파싱된 후 뷰 영역에서 처리할 수 있는 JavaBean 형태의 클래스 객체에 저장되어 전달된다.

④ 처리된 정보를 뷰 영역으로 전달

```
// 모델의 로직을 실행하는 메서드
public void processAction( ... ) throws Exception {
    String address = serverAddress;
    String result = "";
    if ( null == address || "".equals(address) ) {
        setForward(req, "portlet.ext.coportlet.error");
    } else {
        // JavaBean에 처리 결과를 저장
        result = getServerInfoFor(address, name, pass);
        req.setAttribute("result", result);
        setForward(req, "portlet.ext.coportlet.success");
    }
}
```

```
// 정보 처리 후 새로운 뷰 영역 페이지 호출
public ActionForward render(...) ... {
    String reqs = getForward(req);
    if (reqs != null && !reqs.equals("")) {
        return mapping.findForward(getForward(req));
    } else {
        return mapping.findForward(
            "portlet.ext.coportlet.init");
    }
}
```

그림 9. 뷰 영역으로 협업 정보를 전달
Fig 9. Transferring the Information to View

〈그림 9〉는 뷰 영역에서 전달받은 요청에 따라 모델 영역의 비즈니스 로직을 실행하는 processAction 메서드와 모델 영역에서 처리된 데이터를 뷰 영역으로 전달해주는 render 메서드를 보여준다. 비즈니스 로직을 처리하는 서블릿 클래스들 중에 뷰 영역의 요청에 의해 처음으로 실행되는 클래스는 PortletAction 클래스이다. PortletAction 클래스가 실행되면 processAction 메서드가 우선적으로 호출되고 전달 받은 데이터를 바탕으로 요청을 처리하게 된다. 요청된 데이터가 처리되고 나면 render 메서드를 통하여 뷰 영역으로 데이터를 전달하게 된다. render 메서드는 Struts 컨트롤러에 정의된 경로를 참조하기 위하여 ActionMapping 클래스 객체의 findForward 메서드를 사용한다.

3.2.2 협업지원 포틀릿의 뷰 영역 구현

협업지원 포틀릿에서 프레젠테이션 로직을 담당하는 뷰 영역은 처리된 협업 정보를 사용자의 요구에 맞도록 화면을 표현하는 JSP 페이지로 구성된다. 뷰 영역은 협업 정보를 효과적으로 표현하기 위하여 사용자 정보를 나타내는 페이지 영역, 사용자가 접속한 서버의 접근 가능한 작업장의 정보를 트리형태로 나타내는 페이지 영역, 그리고 선택한 작업장의 하위 리소스 정보와 해당 리소스를 제어할 수 있는 기능을 제공해주는 페이지 영역으로 구성된다.

(1) 작업장 트리 페이지 구현

작업장 트리 페이지는 포틀릿의 왼쪽 프레임에서 사용자가 CoSlide 협업시스템 서버에 접속한 뒤 접근할 수 있는 각 작업장들의 정보를 트리 형태로 보여주는 역할을 한다. 사용자가 접근할 수 있는 개인작업장, 공개작업장 그리고 그룹작업장의 리스트를 각각 트리 형태로 표현하여 원하는 작업장을 손쉽게 선택할 수 있다.

```
while(p_ST.hasMoreTokens()){
    StringTokenizer p_String = new
        StringTokenizer((String)p_ST.nextToken(),"/");
    ...
    // 트리 노드에 추가될 작업장 정보 추출
    while(p_String.hasMoreTokens()){
        String p_temp = p_String.nextToken();
        resource = resource + "/" + p_temp;
        Enumeration p_e = p_h.keys();
        while(p_e.hasMoreElements()){
            String p_temp1 = (String)p_e.nextElement();
            if(resource.equals(p_temp1)){
                i = ((Integer)p_h.get(p_temp1)).intValue();
                p_tok = false;
            }
        }
        if(p_tok)
    }
    // 추출한 작업장 정보를 트리 노드에 추가
    d.add((<%= j %>,<%= i %>,<%= p_temp %>');
}
}
```

그림 10. 작업장 트리 페이지 구성
Fig 10. Composing the Workspace Tree Page

〈그림 10〉은 자바스크립트로 구현된 작업장 트리 페이지에서 작업장의 이름을 바탕으로 트리의 노드를 구성하는 부분을 보여준다. 모델 영역에서 문자열 형태로 전달받은 작업장들 리스트는 이름으로 분류하여 트리의 노드로 추가한다. 사용자가 작업장 트리 페이지에서 노드를 선택하면 선택된 노드의 작업장 정보를 바탕으로 작업장 페이지를 호출하게 된다.

(2) 작업장 페이지 구현

작업장 페이지는 포틀릿의 오른쪽 프레임에서 작업장의 세부 정보와 하위 리소스들을 제어할 수 있는 기능을 표현하는 영역이다.

```
</tr>
<logic:iterate name="RL" property="data" id="ud">
<tr>
<td>
...
<td>
// 리소스의 타임 값에 따른 컬렉션/파일 분류
<logic:equal name="ud" property="type" value="F">
    <bean:define id="filedown" >...</bean:define>
    <a href="<bean:write name="filedown" />" ... >
<bean:write name="ud" property="name" /> </a>
</logic:equal>
<logic:equal name="ud" property="type" value="T">
    <bean:define id="folder" >...</bean:define>
```

```

<a href="<bean:write name="folder" />" ... >
<bean:write name="ud" property="name" /></a>
</logic:equal>
</td>
<td>
<bean:write name="ud" property="length" />
</td>
...
</tr>
</logic:iterate>
    
```

그림 11. 작업장 페이지 구성
Fig 11. Composing the Workspace Page

(그림 11)은 뷰 영역의 작업장 페이지에서 Struts 태그 라이브러리와 JavaBean 클래스를 이용하여 작업장의 정보를 표시하는 프레젠테이션 로직을 보여준다. 작업장 페이지에서는 모델 영역에서 전달받은 JavaBean 클래스를 바탕으로 작업장의 정보를 불러오고 이 정보를 바탕으로 Struts 태그 라이브러리를 이용하여 화면을 구성한다. Struts 태그 라이브러리는 복잡한 코드의 기술을 줄이고 HTML 태그와 조화롭게 사용하여 효과적으로 프레젠테이션 로직을 구현하고 비즈니스 로직과 분리하는 역할을 한다.

3.2.3 협업지원 포틀릿의 모델 영역 구현

협업지원 포틀릿에서 컨트롤러 영역은 Struts 구조에 따라 모델과 뷰 영역을 구성하는 파일들의 연결과 데이터의 흐름을 정의하는 역할을 한다. Struts에서 제공하는 ModuleConfig 클래스 객체는 외부에 정의된 XML 문서의 정보를 바탕으로 생성되며, 이 객체는 포틀릿이 실행될 때 요청에 따라 정해진 흐름으로 데이터를 전달하거나 페이지를 호출하는 것을 지원한다. 이러한 구조는 새로운 파일이나 서비스를 추가할 때 외부의 XML을 통하여 새롭게 정의함으로써 모델 영역이나 뷰 영역을 구성하는 파일의 내용이나 구조를 변경시키지 않고 적용할 수 있도록 도와준다.

(1) Struts-Config.xml 파일

Struts-Config.xml 파일은 Struts 구조에 따라 모델 영역과 뷰 영역을 구성하는 파일들과 데이터의 흐름을 정의한다.

```

<action-mappings>
...
// 초기 화면을 정의하는 액션
<action path="/ext/coportlet/init"
        forward="portlet.ext.coportlet.init" />
<action path="/ext/coportlet/server" name="tf"
        type="com.ext.portlet.coportlet.action.Server">
// 액션 처리 결과에 따라 success 액션으로 이동
    
```

```

<forward name="portlet.ext.coportlet.success"
        path="portlet.ext.coportlet.success" />
...
</action>
</action-mappings>
    
```

그림 12. 영역 간 연결과 데이터 흐름의 정의
Fig 12. Definition Links and Data Flow

(그림 12)는 모델 영역과 뷰 영역을 구성하는 파일들을 연결하고 데이터의 흐름을 정의하는 Struts-Config.xml 파일의 일부를 보여준다. 포틀릿에서 새로운 페이지를 호출하거나 데이터를 전달하는 행위를 Action이라고 표현하는데 Action은 Struts-Config.xml에 정의된다. Action을 정의하기 위해서는 <action-mappings> 하위에 <action> 태그를 한다. <action> 태그는 특정 페이지나 서블릿 클래스를 가리키고 있는데 해당 페이지나 서블릿 클래스에서 액션이 발생되어 새로운 연결이나 데이터 흐름이 발생할 때 이를 정의하기 위하여 하위에 <forward> 태그를 정의해야 한다. (그림 12)에서 "/ext/coportlet/server"로 path 속성 값을 정의하고 있는 <action> 노드는 path 속성 값을 바탕으로 서블릿 클래스와 JSP 페이지 등에서 호출될 수 있으며, 이벤트가 발생하면 하위의 <forward> 노드의 name 속성 값에 따라 새로운 페이지를 호출한다.

IV. 협업지원 포틀릿의 동작 및 성능

협업지원 포틀릿은 웹 기반의 포털에서 사용자 중심의 인터페이스를 구성하는 컴포넌트로서 동작한다. 사용자는 자신의 작업 환경을 고려한 맞춤형 인터페이스를 통하여 다양한 서비스와 더불어 협업지원 포틀릿을 사용함으로써 효율적으로 CoSlide 협업시스템을 이용할 수 있다.

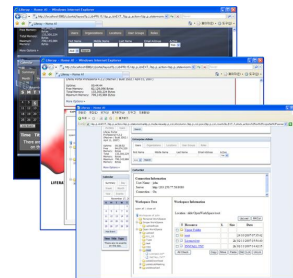


그림 13. 포털에서 동작하는 협업지원 포틀릿

Fig 13. Collaboration Portlet on a Portal

4.3.2 CoSlide 협업시스템과 타 시스템 비교

CoSlide 협업시스템은 기존에 지원하던 여러 협업 클라이언트와 함께 협업지원 포틀릿을 지원함으로써 사용자들의 다양한 작업 환경을 고려한 협업 환경을 구성할 수 있으며, 다양한 플랫폼에 적합한 여러 클라이언트를 제공함으로써 보다 효과적으로 협업 수행을 지원할 수 있다. <표 4>는 기존의 협업시스템과 함께 개발된 협업지원 포틀릿을 지원하는 CoSlide 협업시스템의 차이점을 보여준다.

표 4. 각 협업시스템들의 특징
Table 4. Features of the Collaborative Systems

	BSCW	iPlace	CoSlide
기본 프로토콜	http	http	WebDAV
개인작업장	○	○	○
그룹작업장	△	○	○
공개작업장	×	○	○
서버 구현기술	Phython	EJB+JSP	JAVA
서버 플랫폼	Unix, Windows	Unix, Linux, Windows	Unix, Linux, Windows
윈도우 클라이언트	×	×	○
리눅스 클라이언트	×	×	○
웹 기반 클라이언트	○	○	○

BSCW[19]는 대표적인 웹 기반의 협업시스템으로서 사용자들을 위한 개인작업장과 그룹 사용자들을 위한 그룹작업장을 부분적으로 지원한다. 하지만 다양한 작업 환경을 고려한 클라이언트와 기능을 지원하지 않고 복잡한 가상공간을 지원하지 않아 효과적인 협업서비스를 제공하지 못한다. iPlace[20]는 웹 기반의 협업시스템으로서 다양한 가상공간을 지원하여 효과적인 협업 수행을 지원한다. 하지만 http 프로토콜의 제약에 따라 가상공간 관리나 자원 제어를 하기 위한 새로운 기능을 추가하기 어려우며 다양한 클라이언트를 지원하지 못한다. CoSlide 협업시스템은 기존의 협업시스템에서 제공하던 가상공간을 사용자와 그룹을 위하여 세분화하였으며, 각 작업장에서 협업 수행을 위한 다양한 기능들을 제공한다. 또한, 사용자

의 작업 환경을 고려한 다양한 플랫폼의 클라이언트를 지원하여 보다 효과적으로 협업을 수행할 수 있도록 도와주며 사용자 중심의 맞춤형 인터페이스를 제공하는 포틀릿 클라이언트를 통하여 그 활용 범위를 더욱 넓히고 있다.

V. 결론

본 논문은 CoSlide 협업시스템을 이용하여 다양한 사용자들의 협업 수행을 지원하기 위한 협업지원 포틀릿의 개발에 대하여 기술하였다. Struts 프레임워크를 기반으로 설계된 협업지원 포틀릿은 MVC 모델링 구조를 바탕으로 협업 정보를 표현하는 사용자 인터페이스를 구현하는 뷰 영역과 CoSlide 협업시스템 서버에 연결하고 협업 정보를 처리하기 위한 WebDAV 메서드를 요청하는 모델 영역으로 구현되어 있으며, 이러한 각 영역의 유기적인 연결과 정보의 흐름을 정의하기 위하여 컨트롤러 영역이 정의되어 있다. 협업지원 포틀릿은 Struts 프레임워크를 바탕으로 구조적으로 설계되어 있으며, WebDAV API를 이용하여 WebDAV 서버와 연동함으로써 효과적으로 협업 정보를 활용할 수 있다. 개발된 포틀릿은 CoSlide 협업시스템 서버에서 제공하는 다양한 가상공간간의 표현을 지원하며, 각 가상공간의 자원을 제어할 수 있는 여러 기능들을 제공한다. 또한, 포탈에서 제공하는 다양한 포틀릿과 함께 서비스되어 사용자의 작업 환경을 고려한 맞춤형 인터페이스를 구성하며, 보다 효과적으로 협업을 수행할 수 있는 환경을 제공한다.

본 논문에서는 Struts 프레임워크를 기반으로 하고 WebDAV 프로토콜을 통하여 CoSlide 협업시스템과 연동하여 협업 수행 기능을 제공하는 협업지원 포틀릿에 대하여 기술하였다. 그러나 이러한 포틀릿의 개발을 체계적으로 도와주는 별도의 개발도구가 존재하지 않아서 구현에 상당한 어려움이 있었다. 향후, 본 논문에서 기술한 협업지원 포틀릿의 개발 경험을 바탕으로 하여 이러한 포틀릿 전용의 개발도구의 기능을 명세하고 이를 구현하는 연구를 수행할 예정이다.

참고문헌

- [1] Janus Boye, "Portal Software: Passing Fad or Real Value," CMS Watch, 2005.
- [2] Eric Knorr, "The new enterprise portal," InfoWorld, 2004
- [3] Portlet, "http://www.jcp.org/en/jsr/detail?id=168"
- [4] Jakarta Slide, "http://jakarta.apache.org/slide," Apache Projects.
- [5] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, "HTTP Extensions for Distributed Authoring - WEBDAV," RFC 2518, Standards Track, 1999.
- [6] C. Kaler, J. Amsden, G. Celmm, B. Cragen, D. Durand, B. Sergeant, E. Whitehead, "Versioning extensions to WebDAV," IETF Internet Draft, 1999.
- [7] E. James Whitehead, Jr., Meredith Wiggings, "WEBDAV: IETF Standard for Collaborative Authoring on the Web," IEEE Internet Computing, pp. 34-40, 1998.
- [8] 김동호, 박진호, 신원준, 이명준, "웹데브 기반의 효과적인 협업 작업 지원," 한국정보과학회, 한국컴퓨터종합학술대회 2006.
- [9] 김동호, 신원준, 박진호, 이명준, "웹데브 기반의 그룹 작업공간 지원," 한국정보처리학회, 정보처리학회논문지 C 제13-C권, 2006년
- [10] 박진호, 신원준, 김동호, 이명준, "WebDAV 기반의 리눅스 협업시스템 클라이언트," 한국정보과학회, 한국컴퓨터종합학술대회 2006.
- [11] Won-Joon Shin, Dong-Ho Kim, Myung-Joon Lee, "DAView: A Linux WebDAV Client Supporting Effective Distributed Authoring," Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work, GROUP'05, 2005
- [12] Struts, Apache Project, "http://struts.apache.org"
- [13] Sun Java System Portal Server, "http://www.sun.com/software/products/portal_srvr"
- [14] IBM WebSphere Application Server, "http://www-306.ibm.com/software/websphere"
- [15] Oracle Application Server 10g, "http://www.oracle.com/technology/products/ias/portal"
- [16] Liferay, "http://www.liferay.com"
- [17] JSR-168, the Java Portlet Specification version 1.0, p19, Figure 4-1
- [18] JSR-168, the Java Portlet Spectification version 1.0, p20, Figure 4-2
- [19] W. Appelt, P. Mambrey, "Experiences with the BSCW Shared Workspace System as the Backbone of a Virtual Learning Environment for Students," Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 99, Seattle, 1999.
- [20] 안건태, 정명희, 이근웅, 문남두, 이명준, "iPlace: EJB 기술을 이용한 웹 기반 협업시스템," 한국정보처리학회논문지, 제8-D권, pp. 735-746, 2001.

저 자 소 개



이 흥 창

2006년 : 울산대학교 컴퓨터정보통신
공학부 졸업(공학사)
2008년 : 울산대학교 컴퓨터정보통신
공학부 졸업(석사)
2008년 ~ 현재 : 울산대학교 컴퓨터정
보통신공학부 박사 과정
관심분야 : 웹기반 정보시스템, 협업시
스템, 웹기반 분산시스템.



이 명 준

1980년 : 서울대학교 수학과 졸업(학사)
1982년 : 한국과학기술원 전산학과 졸
업(석사)
1991년 : 한국과학기술원 전산학과 졸
업(박사)
1982년 ~ 현재 : 울산대학교 컴퓨터정
보통신공학부 교수
1993년 ~ 1994년 : 미국 버지니아대학
교환 교수
2005년 ~ 2006년 : 미국 캘리포니아주
립대학 교환 교수
관심분야 : 웹기반 정보시스템, 프로그
래밍언어, 분산 프로그래
밍, 생물정보학, 센서네트
워크 프로그래밍 환경