

# 개인화된 서비스를 지원하는 협업지원 포틀릿 개발

이흥창<sup>0\*</sup>, 김성훈\*, 이태호\*, 이명준\*\*

울산대학교 컴퓨터·정보통신공학부

<sup>0\*</sup>myhyunii, \*{heinz, soulfree}@mail.ulsan.ac.kr, \*\*mjlee@ulsan.ac.kr

## Development of a Collaborative Portlet Supporting Individualized Services

Hong-Chang Lee<sup>0\*</sup>, Seonghune Kim\*, Tae-Ho Lee\*, Myungjoon Lee\*\*

School of Computer Engineering & Information Technology, University of Ulsan

### 요 약

인터넷이 발달함에 따라 인터넷 기반의 협업시스템을 이용하여 정보를 교환하고 협업을 수행하는 사용자와 그룹이 급속히 증가하고 있다. 이러한 사용자들이 급속히 증가함에 따라 사용자들의 다양한 작업 환경을 고려한 개인화된 협업서비스가 제공되는 것이 바람직하지만 현재의 협업시스템들은 이러한 서비스를 제공하지 않아서 효과적인 협업 환경을 제공하기 어렵다.

본 논문은 협업시스템을 이용하여 다양한 사용자들의 협업 수행을 지원하는 CoPortlet의 개발에 대하여 기술한다. CoPortlet은 포탈에 포함되는 포틀릿 컴포넌트로서 협업시스템 서버에서 제공하는 다양한 가상공간을 지원하며, 각 가상공간의 자원을 제어하는 기능들을 제공한다. CoPortlet은 Struts 프레임워크를 바탕으로 설계하였으며, WebDAV 프로토콜을 이용하여 협업시스템과 상호 동작한다. 사용자는 여러 포틀릿과 함께 CoPortlet으로 구성되는 포탈을 이용하여 자신의 작업 환경을 고려한 맞춤형 인터페이스를 구성하고 효과적으로 협업을 수행할 수 있다.

### 1. 서 론

인터넷이 발달함에 따라 인터넷을 이용하여 정보를 공유하는 사용자들이 급속하게 늘고 있다. 그에 따라, 효과적으로 정보를 공유하고 협업을 수행할 수 있도록 도와주는 인터넷 기반의 협업시스템이 널리 사용되고 있다. 또한, 사용자들이 급속하게 늘어남에 따라 사용자를 고려한 개인화된 서비스에 대한 요구가 점차 늘고 있다. 하지만 기존의 협업시스템은 이러한 개인화된 서비스를 제공하지 않아서 효과적인 협업 환경을 제공하기 어렵다.

CoSlide[1]는 Jakarta Slide를 확장한 협업시스템으로서 웹 기반으로 협업서비스를 제공하며, 다양한 사용자와 그룹을 지원하기 위하여 개인작업장, 공개작업장, 그룹작업장을 제공한다. CoSlide는 WebDAV[2] 프로토콜을 이용하여 협업클라이언트와 상호 동작하며, 사용자와 그룹을 체계적으로 지원하는 가상공간을 통하여 효과적인 협업 수행 기능을 제공한다.

포탈[3]은 웹 기반의 어플리케이션으로서 다양한 인터페이스를 통하여 개인화된 서비스를 제공하며, 다양한 시스템의 자원을 통합하여 관리하고 제공한다. 포탈은 다양한 사용자들을 고려한 사용자 중심의 인터페이스를 제공하기 위하여 포틀릿[4] 컴포넌트를 사용한다. 포틀릿은 조합을 통하여 웹 화면을 구성하는 웹 컴포넌트로서 포탈은 여러 포틀릿을 조합하여 사용자를 고려한 인터페이스를 구성한다.

본 논문은 협업시스템을 이용하여 다양한 사용자들의 협업 수행을 지원하는 CoPortlet의 개발에 대하여 기술한다. CoPortlet은 포탈에서 동작하는 포틀릿으로서 협업

가상공간과 자원을 효과적으로 제어할 수 있는 기능을 제공한다. CoPortlet은 Struts[5] 프레임워크를 바탕으로 설계되어 비즈니스와 프레젠테이션 영역이 독립적으로 구하였으며, WebDAV 프로토콜을 통하여 협업시스템과 상호 동작한다. 개발된 CoPortlet은 포탈에 포함되어 사용자의 작업 환경을 고려한 다양한 인터페이스를 구성하여 효과적인 협업 환경을 제공한다. 사용자는 포탈을 통하여 다양한 포틀릿과 함께 CoPortlet으로 구성된 사용자 중심의 인터페이스를 바탕으로 자신의 작업 환경에 부합하는 협업 환경을 구성할 수 있다.

본 논문의 구성은 다음과 같다. 1장 서론에 이어 2장에서는 CoPortlet에 관련된 연구를 살펴보고 3장에서는 CoPortlet을 구현하기 위한 설계와 기법에 대하여 살펴본다. 그리고 4장에서는 개발된 CoPortlet의 동작 모습과 평가를 살펴보도록 하고 마지막으로 5장에서는 결론을 통하여 본 연구의 결과를 살펴보도록 한다.

### 2. 관련 연구

본 장에서는 CoPortlet을 개발하기 위한 배경이 되는 포탈, 포틀릿, Struts 프레임워크, CoSlide 협업시스템에 대하여 기술한다.

#### 2.1 포탈(Portal)

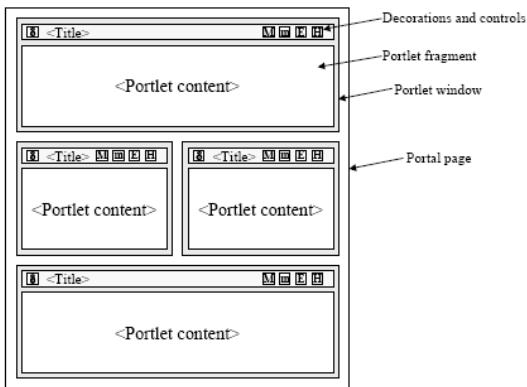
포탈은 개인이나 그룹이 여러 시스템으로 나뉘어 관리해온 정보들을 통합하여 개인이나 그룹의 활동에 필요한 모든 자원을 하나의 웹 화면으로 제공하고 관리, 분석할 수 있는 시스템이다. 포탈은 웹 기반의 게이트웨이로서

\*본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성 지원사업의 연구결과로 수행되었습니다.(IITA-2008-(C1090-0801-0039))

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 포틀릿(Portlet)이라는 컴포넌트를 사용한다 이러한 포틀릿을 활용하여 사용자들을 고려한 다양한 서비스를 제공하는 포탈 서버로는 Sun의 Portal Server[6], Oracle의 OracleAS[7] 등이 있고 이와 함께 오픈 소스 서버인 Liferay Portal[8]이 널리 이용되고 있다

## 2.2 포틀릿(Portlet)

포틀릿은 웹 화면에서 동적인 컴포넌트 구성을 지원하는 환경으로 정적인 인터넷 환경에서 개인화된 서비스를 제공할 수 있는 핵심 기술이다 포틀릿은 하나의 화면을 구성하기 위한 단위가 아닌 화면의 한 요소를 정의하기 위한 마크업 단위이다 포틀릿으로 구성된 웹 화면은 다양한 포틀릿의 조합을 통하여 동일한 요청에도 사용자 별로 다양한 인터페이스를 구성하여 제공할 수 있다 [그림 1]은 하나의 웹 페이지를 구성하는 다양한 포틀릿의 조합을 보여준다



[그림 2] 웹 페이지를 구성하는 포틀릿 조합

## 2.3 Struts

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 1]는 Struts를 구성하는 각 영역을 보여준다

[표 1] Struts의 MVC 모델 구성요소

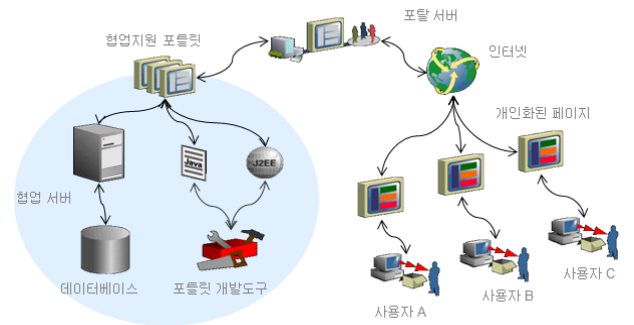
Model	비즈니스 로직 데이터 처리 및 DB 연결
View	프레젠테이션 로직 처리된 데이터를 표현 인터페이스를 구성하여 사용자에게 제공
Controller	모델 과 뷰 영역의 연결을 정의

## 2.4 CoSlide 협업시스템

CoSlide 협업시스템은 WebDAV 프로토콜을 사용하여 사용자와 그룹을 위한 가상공간을 지원하여 효과적으로 협업서비스를 제공하는 협업시스템이다 CoSlide 협업시스템은 개인 사용자를 위한 가상공간인 개인작업장다 양한 사용자가 접근하여 공개된 작업을 수행할 수 있는 가상공간인 공개작업장 특정 그룹에 소속된 사용자들 간의 효율적인 그룹 작업을 지원하는 그룹작업장을 제공한다. CoSlide 협업시스템은 Jakarta Slide를 확장하여 개발된 CoSlide 서버와 CoSlide 서버에서 제공되는 협업 서비스를 활용하는 전용 클라이언트로 구성된다 이러한 전용 클라이언트로는 윈도우즈 클라이언트인 CoSpace[9]와 리눅스 클라이언트인 CoDAView[10]가 개발되어 있다.

## 3. CoPortlet의 개발

본 장에서는 개인화된 협업서비스를 제공하기 위하여 사용자 중심의 인터페이스를 지원하는 CoPortlet의 개발에 대하여 기술한다 [그림 2]는 개발된 CoPortlet을 통하여 다양한 사용자들에게 개인화된 협업서비스를 제공하는 과정을 보여준다



[그림 3] 포탈을 이용한 개인화된 협업서비스 제공

## 3.1 CoPortlet의 설계

### 3.1.1 CoPortlet의 구조

CoPortlet은 협업정보를 효과적으로 처리하고 그 결과를 사용자에게 제공하기 위하여 모델과 뷰 영역을 분리하여 개발하였고 각 영역의 연결 및 데이터 흐름을 유기적으로 정의하기 위하여 컨트롤러 영역을 구현하였다

#### (1) CoPortlet의 뷰 영역

CoPortlet의 뷰는 협업정보를 표현하는 프레젠테이션 로직이 정의되어있는 영역으로서 사용자가 접근할 수 있는 가상공간의 정보를 나타내는 작업장트리 부분과 선택한 작업장 하위의 자원 정보를 보여주고 자원 제어 기능을 제공하는 작업장 부분으로 구분된다 작업장트리 부

분은 모델 영역에서 처리된 협업 정보를 바탕으로 사용자가 접근할 수 있는 가상공간을 트리 형태로 표현한다. 작업장 부분은 사용자가 작업장트리에서 선택한 작업장의 하위 자원 정보와 자원 제어 기능을 제공하며 사용자가 요청한 기능을 모델 영역으로 전달한다.

(2) CoPortlet의 모델 영역

CoPortlet의 모델은 협업 서버와 연결하고 협업 정보를 처리하는 비즈니스 로직이 정의되어있는 영역이다. 모델 영역은 뷰 영역의 요청을 처리하여 서버로 전달하거나 서버로부터 받은 협업 정보를 처리하여 뷰 영역으로 전달하는 역할을 한다.

(3) CoPortlet의 컨트롤러 영역

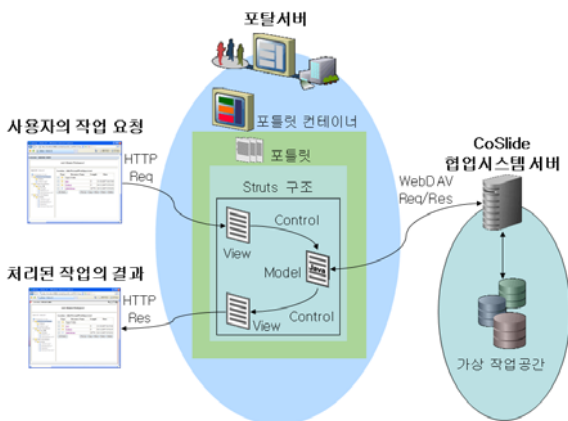
CoPortlet의 컨트롤러는 모델과 뷰 영역의 연결과 데이터의 흐름을 정의하는 영역이다. Struts 프레임워크의 구조에 따라 컨트롤러 영역은 별도의 XML 문서를 바탕으로 정의하며 이는 추후 서비스의 확장이나 영역 간 연결의 변경이 있을 때 코드의 내부를 수정할 필요 없이 손쉽게 적용할 수 있도록 도와준다.

3.1.2 협업시스템과의 연동

CoPortlet은 CoSlide 협업시스템과의 연동을 위하여 DAVExplorer[11]의 WebDAV 라이브러리를 사용한다. 이 라이브러리는 HTTPClient 라이브러리를 상속받아 자바 기반으로 WebDAV 메소드를 API 형태로 제공한다.

3.2 CoPortlet의 구현

CoPortlet은 포탈에 포함되어 사용자 중심의 인터페이스를 구성하여 개인화된 협업서비스를 제공하고 효과적으로 협업시스템과 연동하기 위하여 Struts 프레임워크 구조로 설계에 따라 각 영역의 세부 구현이 필요하다. [그림 3]은 포탈에 포함되어 협업시스템과 연동하며 사용자의 요청을 처리하여 서비스를 제공하는 CoPortlet의 내부 동작을 보여준다.



[그림 4] CoPortlet의 내부 동작

3.2.1 CoPortlet의 모델 영역 구현

(1) 협업시스템 사용자 인증 및 정보 요청

```
import edu.uci.ics.DAVExplorer.*;
...
public String getServerInfor(...){
    WebDAVConnection ws = new WebDAVConnection();
    NVPair[] Headers = new NVPair[3];
    Headers[0] = new NVPair("Host", serverAdd);
    ...
    HTTPResponse h = ws.PropFind("slide/..", c, Headers);
    h.getData();
}
....
```

[그림 5] 협업 서버를 이용하는 모델 영역

[그림 4]는 모델 영역에서 DAVExplorer 라이브러리를 사용하여 협업시스템에 접근하는 코드를 보여준다. CoPortlet은 WebDAVConnection 클래스를 이용하여 협업 서버에 접속하고 이 클래스의 PropFind 메소드를 이용하여 협업시스템의 정보를 요청한다.

(2) XML 정보의 변환

```
NodeList nodeResponse =
    doc.getElementsByTagName("D:response");
for(int i=0;i<nodeResponse.getLength();i++){
    NodeList nodeResponseChild =
        nodeResponse.item(i).getChildNodes();
    uridata = new URIData();
    for(int j=0;j<nodeResponseChild.getLength();j++){
        Node node = nodeResponseChild.item(j);
        if(node.getNodeName().equals("D:href")){
            NodeList nodeHref = node.getChildNodes();
            for(int k=0;k<nodeHref.getLength();k++){
                if(nodeHref.item(k).getNodeType() ==
                    Node.TEXT_NODE)
                    uridata.setURI(nodeHref.item(k).getNodeValue());
            }
        }
    }
    ...
}
```

[그림 6] XML 형태의 정보를 변환

[그림 5]는 협업 서버로부터 받은 XML 형태의 정보를 뷰 영역에서 처리할 수 있는 클래스 형태의 정보로 변환하는 코드를 보여준다. 협업 서버는 클라이언트의 요청에 따라 처리된 결과를 XML 형태로 구성하여 응답하며, 이 정보는 트리 형태로 구성되어 사용자와 그룹에 관한 정보를 효과적으로 정의한다. 서버로부터 받은 XML 정보를 파싱하기 위하여 DOM 파서를 사용하였으며, CoPortlet은 뷰 영역으로 전달하기 위하여 JavaBean 클래스에 변환된 정보를 저장한다.

(2) 뷰 영역으로 데이터 전달

```

public void processAction(...) throws Exception {
    if(null == serverAddress || "".equals(serverAddress)){
        setForward(req, "portlet.ext.coportlet.error");
    } else {
        String result = getServerInfor(serverAddress, ...);
        req.setAttribute("result", result);
        setForward(req, "portlet.ext.coportlet.success");
    }
}

public ActionForward render(...) throws Exception {
    if(getForward(req) != null && !getForward(req).equals("")){
        return mapping.findForward(getForward(req));
    } else {
        return mapping.findForward("portlet.ext.coportlet.init");
    }
}
    
```

[그림 7] 처리된 결과를 뷰 영역으로 전달

[그림 6]는 뷰 영역에서 전달받은 요청에 따라 모델 영역의 비즈니스로직을 실행하는 processAction 메소드와 모델 영역에서 처리된 정보를 뷰 영역으로 전달해주는 render 메소드를 보여준다. 모델 영역은 뷰 영역의 요청에 따라 processAction 메소드를 우선적으로 실행하고, 매개변수로 전달받은 정보를 바탕으로 뷰 영역의 요청을 처리한다. 요청의 내용에 따라 협업시스템 서버에 접근하여 협업 정보를 요청하고 그 결과를 받아 처리하게 된다. 처리된 정보를 render 메소드를 통하여 뷰 영역으로 전달하는데 적절한 뷰 영역을 선택하기 위하여 ActioMapping 클래스를 이용하여 Struts 프레임워크의 컨트롤러 영역을 참조한다.

### 3.2.2 CoPortlet의 뷰 영역 구현

```

...
<logic:iterate name="RL" property="data" id="ud">
...
<td>
<logic:equal name="ud" property="type" value="false" >
<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="true" >
<bean:define id="folder" >...</bean:define>
<a href="<bean:write name="folder" />" target="_self">
    <bean:write name="ud" property="name" /></a>
...
    
```

[그림 8] 협업 정보를 표현하는 작업장 페이지

[그림 7]은 뷰 영역의 작업장 페이지에서 Struts 태그 라이브러리와 JavaBean 클래스를 이용하여 협업 서버의 작업장 정보를 표현하는 프레젠테이션 로직을 보여준다. 뷰 영역의 작업장 페이지에서는 모델 영역에서 전달받은

JavaBean 클래스를 바탕으로 작업장의 기본적인 정보를 호출하고 이 정보를 바탕으로 Struts 태그라이브러리를 이용하여 화면을 구성한다

### 3.2.3 CoPortlet의 컨트롤러 영역 구현

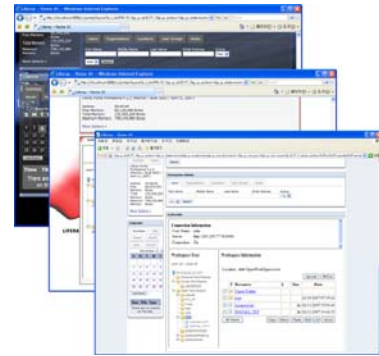
```

<struts-config>
...
<action-mappings>
    <action path="/ext/coportlet/init"
        forward="portlet.ext.coportlet.init" />
    <action path="/ext/coportlet/server" name="tf"
        type="com.ext.portlet.coportlet.action.Server">
        <forward name="portlet.ext.coportlet.success"
            path="portlet.ext.coportlet.success" />
    </action>
...
    
```

[그림 9] 영역 간의 연결을 정의하는 Struts-Config.xml

[그림 8]은 모델과 뷰 영역을 구성하는 파일들의 연결과 데이터의 흐름을 정의하는 Struts-Config.xml 파일의 일부를 보여준다. Struts-Config.xml은 <action> 태그를 통하여 모델과 뷰 영역을 구성하는 파일들을 정의하고 <forward> 태그를 이용하여 액션이 발생하거나 요청이 발생했을 때 호출하게 될 페이지를 정의한다

## 4. CoPortlet의 동작 및 평가



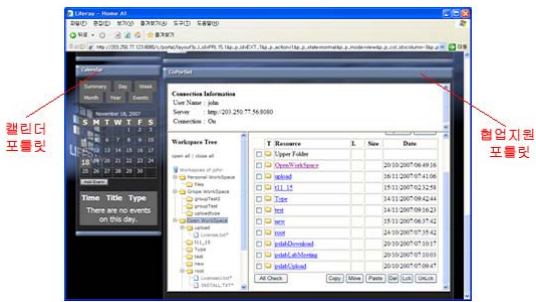
[그림 10] 다양한 인터페이스를 구성하는 CoPortlet

CoPortlet은 웹 기반으로 동작하는 포털에 포함되어 사용자 중심의 인터페이스를 구성한다. 포털은 다양한 포틀릿들을 조합하여 사용자별로 인터페이스를 구성하며, 사용자는 자신의 작업 환경을 고려한 인터페이스를 선택하고 다양한 서비스를 효율적으로 제공받을 수 있다. CoPortlet은 다른 포틀릿들과 함께 사용자의 환경을 고려한 인터페이스를 구성함으로써 효율적인 협업 환경을 제공할 수 있다. [그림 9]는 CoPortlet이 다양한 포틀릿과 함께 포털에서 제공하는 인터페이스를 통하여 복합적으로 제공되는 화면을 보여준다

### 4.1 CoPortlet의 동작



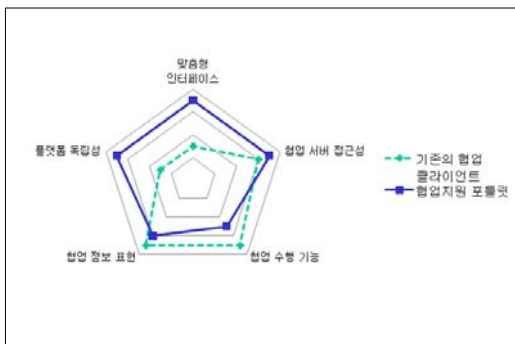
CoPortlet은 협업 서버에 접속하여 협업을 수행하는 사용자 및 서버의 기본 정보를 표현하며 사용자가 접근 가능한 사용자별 개인작업장 그룹작업장, 공개작업장을 트리 형태로 표현한다 트리 형태로 표현된 작업장 목록에서 작업하기를 원하는 작업장을 선택하면 그 작업장의 세부 정보가 메인 화면에 나타나게 된다 CoPortlet은 선택한 작업장의 하위 자원들과 폴더의 정보를 리스트 형태로 표현하며, 각 자원 및 폴더를 제어할 수 있는 기능을 제공한다. 사용자는 원하는 작업장을 선택하고 파일 업로드, 복사, 이동, 잠금 등을 활용하면서 다른 사용자와 함께 손쉽게 협업을 수행할 수 있다. [그림 10]은 포털에서 동작하는 CoPortlet의 모습을 보여준다. CoPortlet은 다른 포틀릿들과 함께 포털의 페이지를 구성하며, 사용자는 다양한 포틀릿과 함께 제공되는 CoPortlet을 통하여 별도의 추가 작업 없이 손쉽게 협업 환경을 구성할 수 있다.



[그림 11] 포털에서 동작하는 CoPortlet

## 4.2 CoPortlet의 평가

CoPortlet은 기존의 협업 클라이언트와 비하여 다양한 장점을 가진다. CoPortlet은 웹을 기반 한 포털에서 동작함으로써 사용자의 작업 플랫폼에 독립적으로 서비스될 수 있고 사용자는 협업시스템을 이용하기 위하여 별도의 클라이언트를 설치할 필요가 없다 또한, 다른 포틀릿과 함께 사용자의 작업 환경을 고려한 맞춤형 인터페이스를 구성함으로써 사용자가 손쉽게 사용할 수 있다 그러나 CoPortlet은 웹 기반으로 구현됨에 따라 상대적으로 협업 수행 기능에 제약이 따르고 협업정보를 표현함에 있어 보다 능동적으로 대처하기 어려운 점이 있다



[그림 12] 기존의 협업클라이언트와 CoPortlet

[그림 11]은 기존의 협업클라이언트와 비교하여 CoPortlet이 가지는 특징을 보여준다

## 5. 결론

본 논문은 협업시스템을 이용하여 다양한 사용자들의 협업 수행을 지원하기 위한 CoPortlet의 개발에 대하여 기술하였다. CoPortlet은 포털에서 동작하는 포틀릿으로서 Struts 프레임워크를 기반으로 구현하였다 CoPortlet은 Struts 프레임워크를 바탕으로 협업 정보를 표현하고 Struts 태그라이브러리와 JSP 페이지를 통하여 사용자 인터페이스를 제공하는 뷰 영역 협업시스템 서버에 접근하고 사용자의 요청에 따라 협업 정보를 처리하는 모델 영역, 뷰와 모델 영역 간의 연결과 데이터 흐름을 정의하는 컨트롤러 영역으로 구현하였다 CoPortlet은 WebDAV 프로토콜을 이용하여 협업시스템 서버와 연결하여 협업을 위한 효과적인 기능을 지원한다

본 논문에서는 Struts 프레임워크를 기반으로 하고 WebDAV 프로토콜을 통하여 협업시스템과 연동하여 협업 수행 기능을 제공하는 CoPortlet의 개발에 대하여 기술하였다. CoPortlet은 포털에 포함되어 사용자의 작업 환경을 고려한 맞춤형 인터페이스를 구성하며 다양한 가상공간을 지원하여 보다 효과적으로 협업을 수행할 수 있는 환경을 구성한다

## 6. 참고문헌

- [1] Dongho Kim, Wonjoon Shin, Jinho Park, Myungjoon Lee, "Supporting Group Workspaces Based on WebDAV," Korea Information Processing Society, 2006.
- [2] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, "HTTP Extensions for Distributed Authoring - WEBDAV," RFC 2518, Standards Track, 1999.
- [3] Janus Boye, "Portal Software: Passing Fad or Real Value," CMS Watch, 2005.
- [4] Portlet, "http://www.jcp.org/en/jsr/detail?id=168"
- [5] Struts, Apache Project, "http://struts.apache.org"
- [6] Sun Java System Portal Server, "http://www.sun.com/software/products/portal\_srvr"
- [7] Oracle Application Server 10g, "http://www.oracle.com/technology/products/ias/portal"
- [8] Liferay, "http://www.liferay.com"
- [9] Dongho Kim, Jinho Park, Wonjoon Shin, Myungjoon Lee, "Supporting Effective Collaborative Works Based on WebDAV," Korea Information Science Society, Vol.33, No. 2(D), pp.566~570, 2006.
- [10] Jinho Park, Wonjun Shin, Dongho Kim, Myungjoon Lee, "A WebDAV-based Collaborative System on Linux," Korea Information Science Society, Vol.33, No. 2(D), pp.611~615, 2006.
- [11] DAVExplorer, "http://www.davexplorer.org"