

사용자 속성 관리의 효율적 지원을 위한 WebDAV 프로토콜의 확장

정혜영[†] · 김동호^{**} · 안건태^{***} · 이명준^{****}

요약

WebDAV(Web-based Distributed Authoring and Versioning)는 웹 기반의 분산 저작과 버전관리를 지원하는 프로토콜로서 인터넷을 통하여 다양한 콘텐츠의 비동기적인 협업을 지원하는 표준 하부구조를 제공한다. WebDAV의 속성 관리의 자원의 주요 정보를 속성으로 설정하여 관리할 수 있는 기능이며, 이들 중 사용자 속성은 사용자 측에서 자유롭게 정의되고 관리될 수 있다는 특징이 있다. 이러한 사용자 속성의 자유로운 설정 기능은 협업시스템과 같은 웹 기반의 응용 시스템을 WebDAV 기반으로 개발하는데 매우 유용하다. 그러나 기존의 WebDAV 속성 관리 기능만으로는 다양한 응용 시스템 개발에 한계가 있다.

본 논문에서는 WebDAV의 사용자 속성 관리를 효율적으로 지원하기 위하여 기존의 WebDAV를 확장한 *DavUP(WebDAV User property design Protocol)* 프로토콜과 이의 활용에 대하여 기술한다. DavUP은 응용 시스템의 컬렉션 구조와 자원의 속성구조에 대한 정의를 제공하고 있으며, 이를 처리하기 위하여 WebDAV 프로토콜에 새로운 헤더를 추가하고 관련된 WebDAV 메소드의 정의를 확장하였다. DavUP 프로토콜의 실제적인 활용을 위하여 자체 개발한 *DAVinci WebDAV* 서버가 DavUP 프로토콜을 지원하도록 확장하였으며, 확장된 *DAVinci* 서버를 이용하여 사용자에게 효과적인 자료 공유 및 교환 기능을 제공하는 일반적인 공개작업장을 실험적으로 개발하여 DavUP 프로토콜의 유용성을 검증하였다.

키워드 : WebDAV, 속성 관리, DavUP, DAVinci, 공개작업장

Extending a WebDAV Protocol to Efficiently Support the Management of User Properties

Hye-Young Jung[†] · Dong-Ho Kim^{**} · Geon-Tae Ahn^{***} · Myung-Joon Lee^{****}

ABSTRACT

WebDAV(Web-based Distributed Authoring and Versioning), a protocol which supports web-based distributed authoring and versioning, provides a standard infrastructure for asynchronous collaboration on various contents through the Internet. A WebDAV property management is a function to set and manage the main information of the resources as properties, and a user property, one kind of the WebDAV properties, has the ability to be freely defined by users. This free definition of user property makes it very useful to develop web-based applications like a collaboration system based on WebDAV. However, with an existing WebDAV property management scheme, there is a limit to develop various applications.

This paper describes a *DavUP(WebDAV User property design Protocol)* protocol which extended the original WebDAV and its utilization which efficiently supports management of WebDAV user properties. DavUP needs the definition of the collection structure and type definition properties for an application. To do this, we added a new header and appropriated WebDAV method functions to the WebDAV protocol. To show the usefulness of DavUP protocols, we extended our *DAVinci WebDAV* server to support DavUP protocols and experimentally implemented a general Open Workspace, which provides effective functions to share and exchange open data among general users, on the *DAVinci*.

Key Words : WebDAV, Property Management, DavUP, DAVinci, Open Workspace

1. 서론

인터넷의 발달로 인하여 지역적으로 원거리에 위치한 다

수의 작업자 그룹들이 웹을 통하여 공동 작업을 수행할 수 있는 다양한 기술들이 개발되어 왔으며 일반적으로 각자의 고유 프로토콜들을 HTTP 프로토콜에 추가로 정의하여 사용해왔다. 그 결과로 유사 응용 시스템들 간의 상호 운용성이 결여되어 시스템의 확장이나 통합이 어려워지게 되었다[1]. 따라서 협업시스템들 간에 상호 운용을 보장하는 웹 기반의 표준화된 분산 저작과 버전관리의 필요성이 증가되었고 이러한 요구를 충족시키기 위하여 W3C의 IETF 산하의 작업 그

※ 이 논문은 2005년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2005-R05-2004-000-10662-0).

† 정 회 원 : (재)울산산업진흥테크노파크 정밀화학지원센터 선임연구원

** 준 회 원 : 울산대학교 컴퓨터정보통신공학부 석사과정

*** 준 회 원 : 울산대학교 컴퓨터정보통신공학과 공학박사

**** 정 회 원 : 울산대학교 컴퓨터정보통신공학부 교수(교신저자)

논문접수 : 2005년 6월 21일, 심사완료 : 2005년 11월 17일

름에 의해서 WebDAV(Web-based Distributed Authoring and Versioning) 명세가 1999년 2월에 발표되었다. WebDAV는 웹 통신 프로토콜인 HTTP/1.1을 확장한 프로토콜로서 인터넷을 통하여 다양한 콘텐츠의 비동기적인 협업을 지원하기 위한 표준 하부구조를 제공한다. WebDAV의 주요 기능으로는 잠금 관리(Lock Management), 속성 관리(Property Management), 컬렉션(Collection), 이름공간 관리(Namespace Management) 등이 있다[2].

WebDAV 자원의 여러 특성을 표현하는데 사용되는 WebDAV 속성은 시스템 속성과 사용자 속성으로 구분되는데, WebDAV 명세서에서는 이러한 속성을 라이브(Live) 속성과 데드(Dead) 속성으로 구분하여 표현하고 있다. WebDAV에서는 자원을 관리하기 위한 몇 가지 시스템 속성만을 미리 정의하고 있으며 사용자 속성은 서버의 자원을 이용하는 사용자 측에서 자유롭게 정의할 수 있다[3]. 즉, WebDAV 자원에 대한 새로운 사용자 속성의 정의와 그 속성의 관리는 전적으로 사용자측에서 담당한다. 사용자 속성의 자유로운 설정은 협업 시스템과 같은 웹 기반의 응용 시스템을 WebDAV 기반으로 개발하는데 매우 유용하다. WebDAV 사용자 속성의 처리에 대한 이러한 유연성은 서버에 등록되는 자원에 동일하게 설정되어야 하는 사용자 속성의 일관성을 체계적으로 유지하기 힘들게 하여 역설적으로 사용자들이 사용자 속성을 다양하게 활용하여 고급의 응용시스템을 개발하는데 장애가 되고 있다.

본 논문에서는 WebDAV 기반의 응용시스템을 개발하는데 유용한 WebDAV 사용자 속성의 관리를 효율적으로 지원하는 *DavUP(WebDAV User property design Protocol)* 프로토콜과 이의 활용에 대하여 기술한다. DavUP은 응용 시스템의 컬렉션 구조와 형 정의(Type Definition) 속성을 정의하고 있다. 형 정의 속성은 WebDAV 기반의 응용 시스템을 개발하기 위하여 클라이언트 측에 의해서 관리되어지는 사용자 속성을 서버에 설정하기 위한 속성이다. 또한 WebDAV 서버에 다양한 사용자 속성 형을 저장하기 위해서 응용 시스템의 컬렉션 구조를 정의하고 있다. 컬렉션 구조는 형 정의 정보를 조회하거나, 자원이 등록될 때 사용자 속성을 처리하는 일련의 수행 절차를 기술하고 있다. DavUP에서는 정의된 컬렉션 구조와 형 정의 속성을 처리하기 위하여 새로운 헤더를 추가하고 관련된 WebDAV 메소드의 기능을 확장하였다.

DavUP 프로토콜의 실제적인 활용을 위하여 자체적으로 개발한 DAVinci[15] WebDAV 서버를 확장하였다. DAVinci는 아파치 mod_dav 모듈의 서비스 제공자 형태로 구현되어 WebDAV의 자원과 속성 관리를 수행하는데 자원의 콘텐츠는 파일 시스템으로 관리하고 속성 정보는 PostgreSQL로 관리한다. 확장된 DAVinci 서버는 DavUP 프로토콜을 통하여 클라이언트 측에서 관리해야 할 사용자 속성들의 관리를 서버 측에서 수행할 수 있기 때문에 사용자 속성을 활용하는 WebDAV 기반의 응용 시스템 개발을 서버차원에서 지원한다. 개발된 DavUP을 지원하는 DAVinci 서버를 이용하여 다수의 사용자들이 공개된 자료의 공유 및 교환 기능을 제공하

는 공개작업장을 실험적으로 개발하여 DavUP 프로토콜의 유용성을 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서 WebDAV 프로토콜과 WebDAV의 속성에 대하여 소개한다. 3장에서 WebDAV 사용자 속성 관리를 위한 DavUP 프로토콜에 대하여 소개하고 4장에서 DavUP을 지원하는 WebDAV 서버의 구현결과를 기술한다. 5장에서 그 활용 사례를 소개하고 6장에서 결론과 향후 연구 과제를 기술한다.

2. WebDAV

2.1 WebDAV의 소개

WebDAV는 인터넷을 통하여 광범위하고 다양한 콘텐츠의 비동기적인 협업 저작을 지원하기 위한 프로토콜이다. WebDAV는 HTTP/1.1 프로토콜의 확장을 통하여 사용자들에게 원격 서버들의 파일들을 수정하고 관리할 수 있도록 한다. WebDAV의 주요 기능으로는 잠금 관리, 속성 관리, 컬렉션, 이름공간 관리 등이 있으며 기존의 HTTP/1.1의 메소드에 추가적인 메소드를 정의하고 있다[2].

- **잠금 관리**: WebDAV에서는 한 문서에 대하여 한명 이상의 작업자가 동시에 작업할 수 없게 하는 기능을 제공한다. 이것은 한 저작자가 자신의 변경 내용을 서버에 반영한 것을 다른 저작자가 자신의 수정 내용에 반영하지 않고 저장 시킴으로 인하여 변경 내용을 잊어버리는 갱신 분실 문제(Lost Update Problem)를 방지한다.
- **속성 관리**: 속성 관리는 웹 자원에 관한 저자, 마지막 수정 일자 등과 같은 정보에 대한 생성, 삭제 그리고 검색을 할 수 있도록 한다. WebDAV 속성은 이름/값의 쌍으로 이루어져 있다. 속성 이름은 URL이며, 속성 값은 잘 정의된 XML 문서이기 때문에 광범위한 XML 형태의 데이터 저장이 가능하다.
- **컬렉션**: WebDAV는 자원을 저장할 수 있는 컬렉션 개념을 도입하고 있다. WebDAV를 통한 자원 관리에는 컬렉션 자신과 컬렉션 내에 있는 자원이나 파일에 대한 생성, 이동, 복사 그리고 삭제 기능을 포함한다.
- **이름공간 관리**: 이름공간 관리는 서버의 이름공간 내에 있는 웹 문서를 복사하고 이동할 수 있는 능력을 제공한다. 그리고 자원 저장 공간인 컬렉션의 콘텐츠를 생성하고 내용을 보여준다. HTTP는 개별 자원에 대한 직접 접근만을 처리하지만 WebDAV는 데이터를 보다 효율적으로 구성하는 수단을 제공한다.

2.2 WebDAV의 확장

WebDAV 프로토콜의 핵심 기능을 강화하기 위하여, 여러 가지 연관되는 프로토콜들이 정의되고 있다. WebDAV 명세가 가지는 한계를 극복하고 새로운 능력을 확장하기 위하여 DeltaV[4], Access Control[5], DASL[6]과 같은 다양한 연관된 표준화 작업이 진행되었고 또한 추가적인 표준화가 진행되고 있다. 그리고 W3C의 WebDAV와 관련된 표준화 작업

과는 별도로 WebDAV를 응용 시스템에 활용하는 업체들이나 연구 사례도 있다.

- **MS Exchange Server**: 기업 내 정보 협업을 지원하는 시스템으로 기존의 WebDAV의 기능에 추가적으로 Batch나 Notification과 같은 메서드의 추가와 새로운 헤더의 추가를 통하여 일괄 처리나 이벤트정보 인지를 제공한다.
- **WebDAD**: WebDAD는 WebDAV 기반의 데이터베이스 저작을 위한 기반구조를 제공하는 연구 결과로 메서드의 데이터베이스 처리를 위한 Batch와 Search 메서드를 WebDAV에 추가적으로 정의한 연구와 구현 사례를 보여주고 있다.
- **DavSUDP**: DavSUDP는 DavUP의 선행연구로서 WebDAV 서버에 설정되는 클라이언트 측의 사용자 속성을 표현하기 위한 환경 설정 속성의 정의와 서버에 의해서 수행되어야 할 절차를 정의한 단순 프로토콜이다. DavSUDP에서 정의하는 환경 설정 속성 또한 WebDAV 사용자 속성이기 때문에 기본적으로 WebDAV 명세를 따르면서 간단하게 서버의 속성 관리 기능을 확장하여 사용자 속성을 지원하고 있다[7].

WebDAV를 활용하는 응용 시스템을 구분하면 크게 3가지로서 WebDAV 서버와 클라이언트, 그리고 WebDAV를 기반으로 하는 응용 시스템으로 나눌 수 있다. WebDAV 서버는 다양한 형태의 영구 저장공간(Persistent Storage)을 가지고 HTTP/1.1을 통한 WebDAV 프로토콜을 이용하여 해당 자원의 관리를 지원한다[8]. WebDAV 클라이언트는 HTTP/1.1을 통하여 자원의 정보를 요청하고 수정한 후에 다시 해당 서버로 저장할 수 있어야 한다. WebDAV를 지원하는 서버 제품으로는 마이크로소프트사의 IIS 5.0, 아파치(Apache)의 mod_dav[9] 모듈, UCSC의 Catacomb[10], Software AG사의 Tamino Server 등이 있다. WebDAV를 지원하는 클라이언트 제품으로는 마이크로소프트사의 MS-Office 2000, SouthRiver Technology사의 WebDrive, Xyθος사의 WebFile Client, 닷넷기반 WebDAV 탐색기[8] 등이 있다.

현재까지의 WebDAV를 지원하는 응용 시스템들은 WebDAV 서버와 클라이언트가 대부분이다. 그래서 다양한 형태와 플랫폼에서 구동하는 서버와 클라이언트들이 개발되었다. 이와 달리 WebDAV 기반의 응용 시스템은 WebDAV 서버를 자원의 저장소로 이용하면서 자원과 속성 정보를 활용하는 응용 시스템을 의미한다. 관련 사례로는 ECCE[11], HATS[12], Posties[13] 등이 있다. ECCE는 다양한 화학과 관련된 문서와 자료 파일을 저장할 때 ECCE에서 정의한 메타데이터와 연관된 WebDAV 사용자 속성들을 사용하였다. 몇몇 WebDAV 기반의 응용 시스템들이 발표되고 있지만 다양하게 개발되지는 못하고 있다.

응용 시스템의 개발은 새로운 비즈니스 응용 시스템이 가지는 다양한 종류의 데이터 자원에 대한 추가적인 속성 정보, 개인적인 데이터 정보, 데이터들 간의 관계 정보 등의 관리가 필요하다. 이러한 자원에 대한 추가적인 정보는 WebDAV 속성 관리를 통하여 해결이 가능하다. 하지만, 지금까지의

WebDAV 서버와 클라이언트들은 속성 관리에 대한 기능과 서비스 제공이 미약하다. 대부분의 WebDAV 서버와 클라이언트들은 속성 관리보다는 자원 그 자체에 대한 관리와 웹을 통한 자원의 원격 저작 기능에 더 중점을 두고 있다. 즉, 현재의 WebDAV 서버는 추가적인 정보를 위한 속성 관리는 하지 않고 기본적인 속성만을 단순히 저장하고 제공하는 기능을 수행하고 있다. 또한 WebDAV 클라이언트는 이러한 속성 정보를 다양하게 활용하지 못하고 단순히 표시해 주는 기능만을 제공한다. 만약 WebDAV 서버가 자원에 대한 사용자 속성을 효과적으로 관리할 수 있는 기능을 제공하고 클라이언트가 자원에 대한 추가적인 사용자 속성을 쉽게 설계하고 적용할 수 있는 기능을 제공한다면 WebDAV를 이용하는 다양한 응용 시스템을 개발할 수 있을 것이다.

2.3 WebDAV 속성

WebDAV 프로토콜은 모든 웹 자원과 관련하여 기본적으로 몇 개의 속성을 가질 수 있다. WebDAV 속성은 이름과 값의 쌍으로 구성되어 있다. 이름은 이름공간과 속성 이름으로 구성되고, 값은 잘 정형화된 XML 형태이다[14]. WebDAV 속성들은 시스템 속성과 사용자 속성으로 구분된다. WebDAV 명세에서는 이러한 속성을 라이브 속성과 데드 속성으로 구분하여 표현하고 있다.

- **시스템 속성**: WebDAV 자원에 대한 시스템 속성은 서버가 속성의 값을 제공한다. 시스템 속성은 모두 11개가 미리 정의되어 있다. 시스템 속성의 역할은 서버에 의해서 계산된 정보를 저장하거나, 잠금과 같은 상태 정보를 관리하거나, 환경정보 저장 용도로 사용되는 등의 서버가 자원을 관리하는데 필요한 내부적인 정보이다.
- **사용자 속성**: WebDAV 자원에 대한 사용자 속성은 문법, 의미, 그리고 일관성 등이 클라이언트에 의해서 유지된다. 그리고 서버는 데이터에 대한 처리를 거의 수행하지 않는다. 이러한 속성들은 클라이언트 응용 프로그램에 의해서 설정되고 갱신된다.

WebDAV 사용자 속성의 관리는 사용자 속성에 대한 추가, 갱신, 삭제에 대한 모든 관리 책임을 클라이언트 측에서 가진다. 즉, 클라이언트는 WebDAV에 새로운 자원을 추가할 경우, 해당 자원별로 적용되어야 할 사용자 속성을 정의하고 PROPPATCH 메소드를 이용하여 사용자 속성을 설정하는 작업을 수행하여야 한다. 또한 WebDAV 서버에 저장된 자원에 설정된 사용자 속성에 대한 정보를 확인하기 위하여 클라이언트는 PROPFIND 메소드를 이용하여 사용자 속성의 종류와 설정 상황을 확인하게 된다. 클라이언트 입장에서 하나의 자원에 대하여 내용과 속성을 WebDAV 서버에 등록하는 것은 단일 처리에 해당하지만, 실제 WebDAV 서버에 등록하는 절차는 PUT과 PROPPATCH 메소드로 구성된 2단계의 요청 수행을 통해서만 가능하다. 이러한 처리 절차는 WebDAV의 사용자 속성을 활용하는데 있어서 장애 요인으로 작용하고 있다.

이러한 문제를 해결하기 위하여 선행 연구 사례로 제시한

DavSUDP에서는 WebDAV 사용자 속성의 정의를 클라이언트에 의해서 한번만 수행하고 자원의 등록이나 변경에 따른 속성의 설정은 WebDAV 서버에 의해서 수행한다. 즉, 사용자 속성의 정보를 서버의 상위 컬렉션에 정의하고 클라이언트에 의해서 자원이 등록될 때 사용자 속성 또한 서버에 의해서 설정되도록 한다[7]. 그러나 DavSUDP를 기존의 응용 시스템에 적용하여 구현하는 과정에서 다음과 같은 몇 가지 특징을 발견할 수 있다.

- DavSUDP는 컬렉션이나 문서가 등록될 때 하위 자원에 대하여 상위와 동일한 사용자 속성 구조를 가진다. 이것은 하위 자원에 대하여 다양한 사용자 속성의 지정이 불가능하다.
- 데이터 모델에 대하여 다양하고 복잡한 사용자 속성의 정의가 불가능하다. DavSUDP에서 정의한 환경 설정 속성에서는 컬렉션과 자원별로 한 개의 사용자 속성 정의만을 지정할 수 있다.
- 각 컬렉션 내에 등록된 자원별로 별도의 개인적인 사용자 속성의 정의가 어렵다. 예를 들어, 서버에 등록된 자원의 특성에 따라 관리되어야 할 속성이 다를 수 있지만, 상위의 문서에 등록된 속성 정보를 그대로 가지게 된다.
- DavSUDP를 지원하는 WebDAV 서버는 기존 WebDAV 명세를 따르는 서버와의 호환성이 매우 높다. 따라서 DavSUDP를 지원하지 않는 WebDAV 서버와 클라이언트의 경우는 DavSUDP 속성 정보를 단순하게 하나의 사용자 속성으로 인식한다.

이상과 같은 특징을 통해서 DavSUDP는 WebDAV 자원에 대하여 동일한 사용자 속성을 상속하는 응용 시스템에 매우 유용하게 활용이 가능하다. 그러나 현실적으로 일반적인 응용 시스템의 경우, 복잡하고 다양한 데이터 모델과 각각의 데이터 모델에 대한 사용자 속성 또한 다양하게 존재할 수밖에 없다. 이러한 응용 시스템을 DavSUDP 프로토콜로써 지원하는 데는 한계가 있다. 따라서 좀 더 다양한 사용자 속성을 정의하고 적용할 수 있도록 하기 위하여 다음 장에서 DavUP 프로토콜을 제안한다.

3. DavUP: WebDAV 사용자 속성 관리의 효율적 지원을 위한 확장된 프로토콜

3.1 DavUP의 소개

본 장에서 소개하는 *DavUP(WebDAV User property design Protocol)* 프로토콜은 WebDAV 자원에 대한 사용자 속성 집합을 다양한 형(Type)으로 제공하고, WebDAV 서버의 자원에 쉽게 적용할 수 있다. 또한 전체적인 응용 시스템을 컬렉션 구조 형태로 사전에 설계하여 서버에 적용한다. 이러한 기능을 지원하기 위해서는 기존의 WebDAV 명세만으로는 한계가 있어서 표준 WebDAV 명세에 추가적인 헤더 정의와 이를 위한 메소드의 기능을 확장하는 방법을 통하여 해결하였다.

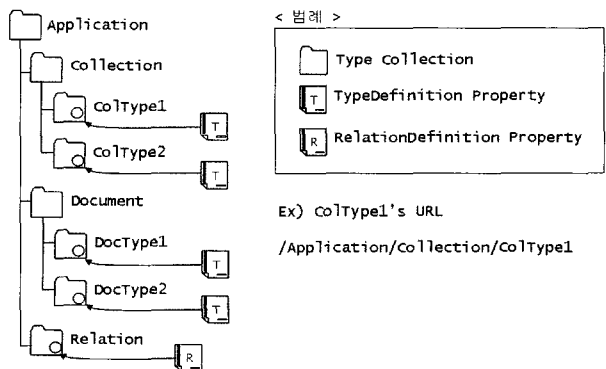
먼저 WebDAV 사용자 속성 관리를 효과적으로 지원하도록 기존의 WebDAV 명세를 확장하는 연구를 수행하는데 있

어서 다음과 같은 기본적인 요구사항을 만족하여야 한다.

- WebDAV 표준 명세를 그대로 수용해야 한다. 기존의 표준을 준수하면서 사용자 속성을 위하여 프로토콜을 확장해야만 기존의 WebDAV 서버가 추가적인 프로토콜을 지원하기 위한 확장 구현이 용이하다.
- WebDAV의 사용자 속성을 활용하려는 사용자가 쉽게 사용자 속성을 정의할 수 있어야 한다. 사용자 속성에 대한 정의에 대한 부담은 일반 사용자들이 사용자 속성을 활용한 응용 시스템을 개발하는데 따른 부담을 줄여준다.
- WebDAV 기반의 응용 시스템을 위하여 사용자가 추가한 사용자 속성의 활용이 용이해야 한다. 기존 WebDAV 클라이언트의 속성 관리 기능은 단순하게 속성을 조회하고 보여주는 기능 위주로 되어 있다. 따라서 사용자 속성의 값이 XML 형태의 길고 복잡한 값일 경우, 그 값에 대한 처리는 사용자가 직접 수행하여야 하는 불편함이 있다.

3.2 DavUP의 구조

WebDAV 서버는 해당 서버에 적용될 응용 시스템의 사용자 속성 타입들에 대한 저장이 반드시 필요하다. 이를 지원하기 위하여 WebDAV 서버에 응용 시스템을 위한 컬렉션 구조를 두고, 여기에 설정될 속성 형 정보들을 저장한다. 이 정보는 DavUP 명세에서 응용 시스템을 위하여 사용하도록 정의하고자 한다. 물론 이렇게 정의된 컬렉션도 WebDAV 자원이기 때문에 DavUP을 지원하는 응용 시스템이 아닌 일반적인 WebDAV 클라이언트의 경우, 단순히 하나의 WebDAV 컬렉션으로 인식하게 된다. 이것은 보안적인 문제가 발생할 수 있지만, 이 문제는 향후 WebDAV 접근 제어(Access Control) 명세를 통하여 해결할 수 있을 것이다. 현재는 DavUP을 지원하는 WebDAV 서버와 클라이언트에서 이 부분을 응용 시스템을 위한 정의 영역으로써 사용하도록 한다는 전제하에서 DavUP을 설명한다.



(그림 1) 응용 시스템을 위한 컬렉션 구조

WebDAV 기반의 복잡한 응용 시스템의 데이터 모델을 적용하기 위해서는 사전에 그 응용 시스템에서 사용할 전체적인 데이터 모델을 WebDAV 서버에 정의하고 있어야 한다. 그렇지 않다면 응용 시스템을 지원하는 클라이언트가 데이터 모델에 대한 처리의 책임을 져야한다. 이것은 다양한 클라이언

트 응용 시스템을 개발하는데 장애요소로 작용할 것이다. 따라서 응용 시스템을 위하여 사전에 정의된 속성이나 형 정보를 WebDAV 서버에 설정하고 클라이언트는 자원을 생성 시에 그 타입에 대한 URI 정보만을 제시함으로써 자원에 대하여 타입에 정의된 사용자 속성들을 설정하도록 하는 방식을 제안한다. WebDAV 서버에 응용 시스템의 데이터 모델을 설정하기 위한 방법은 (그림 1)과 같이 응용 시스템을 위한 속성 정의 형태를 표현하기 위한 컬렉션 구조와 사용자 속성을 설정하는 방법을 보여 주고 있다.

응용 시스템의 데이터 구조를 정의하기 위한 컬렉션에는 컬렉션(Collection)과 문서(Document) 정보를 위한 컬렉션이 있고 이 컬렉션 하위에 각각의 컬렉션과 문서별로 적용할 수 있는 다양한 속성 정의 형을 빈(Empty) 컬렉션으로 생성한다. 응용 시스템을 위한 컬렉션 구조에서 형 정의 컬렉션은 자신의 하위 컬렉션의 생성을 통하여 자신의 사용자 속성 정의를 포함하는 자식 형 정의 컬렉션의 생성을 지원한다. 모든 형 설정을 위한 정보는 WebDAV 컬렉션의 사용자 속성으로 정의한다.

3.3 DavUP의 설계

DavUP에서는 사용자 속성 타입을 정의하는 방법으로 응용 시스템을 위한 컬렉션 구조의 자원에 바로 설정될 수 있는 사용자 속성 형태로 정의하고 있다. 사용자가 정의한 속성 값은 XML 형태로 표현된다. 그러한 요소는 DavUP을 지원하는 WebDAV 서버에서는 DavUP 설정 용도로 적용된다. WebDAV 서버가 DavUP을 지원하는지를 확인하고 또한 자원이 어떠한 타입으로 정의되어 있는지를 조회하기 위하여 새로운 헤더를 정의하고 있다. 새로운 헤더 정보는 기존의 WebDAV 명세에 추가된 것이다.

3.3.1 TypeDefinition 속성

WebDAV 서버에 등록되거나 적용될 컬렉션이나 문서 자원에 대한 사용자 속성 타입을 정의하기 위해서 (그림 2)와 같은 XML의 DTD(Document Type Definition)를 정의하였다. (그림 2)의 DTD는 새로운 WebDAV 자원이 생성될 때 해당 자원에 적용될 사용자 속성을 설정하기 위한 타입을 정의하기 위한 것이다. *TypeName* 요소는 타입의 이름을 표현한다. *KindOfType* 요소는 *TypeDefinition* 요소가 컬렉션에 적용될 타입인지 문서에 적용될 타입인지를 구분하기 위한 요소이다. 또한 *Namespace* 요소는 응용 시스템을 설정하기 위한 컬렉션의 URL을 기본적인 이름공간 정보로 가진다. 이것은 해당 응용 시스템의 유일한 구분자이다. *Description* 요소는 해당 타입의 사용자 속성 집합의 용도나 역할에 대한 설명을 표시한다. 끝으로 *Property* 요소는 해당 자원에 적용될 사용자 속성 목록을 가지고 있다. *Property* 요소의 값은 XML 형식을 따르는 자식 요소 형태로 기술한다. 그리고 자식 요소들은 각각이 하나의 적용되는 WebDAV 자원의 사용자 속성으로 사용된다. 자식 요소는 필요할 경우, 기본값을 가질 수 있다. 기본값은 자원이 생성되고 사용자 속성이 적용

될 때 초기값으로 설정된다. 이렇게 함으로써 자원의 생성과 함께 사용자 속성의 값을 설정하는 PROPPATCH 메소드 작업을 별도로 할 필요가 없다.

```
<!ELEMENT TypeDefinition ( TypeName, KindOfType,
                          Namespace, Description, Property ) >
<!ELEMENT TypeName ( #PCDATA ) >
<!ELEMENT KindOfType ( ColType | DocType ) >
<!ELEMENT ColType EMPTY >
<!ELEMENT DocType EMPTY >
<!ELEMENT Namespace ( #PCDATA ) >
<!ELEMENT Description ( #PCDATA ) >
<!ELEMENT Property ANY >
```

(그림 2) DavUP의 사용자 속성 타입 정의를 위한 DTD

예를 들어 새로운 컬렉션을 생성하려고 할 때 적용될 컬렉션의 타입에 대한 정의가 "/Application/Collection/ColType1" 일 경우, 위의 DTD 형식을 따르는 XML 형태의 속성 내용은 (그림 3)과 같이 설정할 수 있을 것이다.

```
<udp:TypeDefinition xmlns:udp="/Application/Collection/ColType1">
  <udp:TypeName>WorkspaceType</udp:TypeName>
  <udp:KindOfType><ColType/></udp:KindOfType>
  <udp:Namespace>/Application/Collection/ColType1</udp:Namespace>
  <udp:Description/>
  <udp:Property>
    <WorkspaceType>OpenWorkSpace</WorkspaceType>
    <WorkspaceOwner>Jung, Hye-Young</WorkspaceOwner>
    <WorkspaceRole/>
  </udp:Property>
</udp:TypeDefinition>
```

(그림 3) DavUP에서 사용자 속성 타입 정의 예제

(그림 3)의 예제에서 컬렉션을 위한 타입의 이름은 *WorkspaceType*이고 작업장을 위한 컬렉션일 경우 *Property* 요소에 *WorkspaceType*, *WorkspaceOwner*, *WorkspaceRole*의 3가지 사용자 속성이 설정되어야 한다는 것을 의미한다. WebDAV 서버에 새로운 컬렉션을 생성할 때 "/Application/Collection/ColType1" 타입을 적용하면 자동으로 3개의 사용자 속성이 설정된다. 그리고 *WorkspaceType*과 *WorkspaceOwner*는 초기값을 가진다.

3.3.2 DavUP-Type 헤더

DavUP-Type 헤더는 WebDAV 서버가 DavUP을 지원하는지를 판단한다. 또한 자원을 생성하는 MKCOL이나 PUT 메소드를 호출할 때 해당 자원에 설정될 사용자 속성 타입을 지정한다. WebDAV 서버에 자원이 생성될 때 *DavUP-Type* 헤더에 설정된 타입 정보를 위한 URI의 속성 설정 정보를 읽어 해당 자원의 사용자 속성으로 설정한다.

```
DavUP-Type = "DavUP-Type" ":" ( "Supported" | TypeResource )
TypeResource = absoluteURI
```

(그림 4) DavUP-Type 헤더 정의

(그림 4)의 DavUP-Type 헤더의 정의에서 헤더의 값은 "Supported"와 TypeResource로 구분된다. OPTIONS 메소드의 요청을 통해서 WebDAV 서버가 DavUP을 지원할 경우, 응답에서 "DavUP-Type: Supported"라는 응답 헤더를 돌려준다. 그리고 MKCOL이나 PUT과 같은 메소드의 요청 시에 자원에 설정할 타입의 URI 값으로 "DavUP-Type: absoluteURI" 헤더를 함께 넘겨주면 WebDAV 서버는 DavUP-Type 헤더를 해석하여 absoluteURI에 정의된 사용자 속성을 해당 자원이 생성될 때 자동으로 설정한다.

3.3.3 WebDAV 메소드의 확장

기본적으로 WebDAV 자원에 사용자 속성 타입을 지정하기 위한 방법으로 DavUP-Type 헤더를 사용한다. 여기서는 이러한 헤더를 인식하고 적용하기 위한 방법을 설명한다. WebDAV 서버가 DavUP-Type 헤더를 지원하기 위해서는 기존의 WebDAV 메소드의 기능을 확장하여야 한다. 확장이 필요한 메소드로는 OPTIONS, MKCOL, PUT 메소드가 있다. 그 외의 메소드는 별도의 기능 수정을 필요로 하지 않는다. 다음에서 확장된 WebDAV 메소드의 기능과 역할에 대하여 상세히 설명한다.

(1) OPTIONS 메소드

OPTIONS 메소드는 웹 서버의 해당 자원에 대한 특성 정보를 요청하는 메소드로 (그림 5)와 같이 요청을 수행한다. DavUP을 지원하는 WebDAV 서버인 경우에는 (그림 6)과 같이 OPTIONS 메소드 요청에 대한 응답에서 DavUP-Type에 대한 정보를 포함하여야 한다. 이 DavUP-Type 정보를 통하여 해당 WebDAV 서버가 DavUP을 지원하는지를 확인할 수 있다. 응답의 DavUP-Type 헤더 값이 "Supported"이면 해당 WebDAV 서버가 DavUP을 지원한다는 것을 의미한다.

```
OPTIONS /index.html HTTP/1.1
Host: www.webdav.or.kr
```

(그림 5) OPTIONS 메소드의 요청 예제

```
HTTP/1.1 200 OK
Server: Apache/1.3.14 DAV/2 Davinci(0.9)
DavUP-Type: Supported
...
```

(그림 6) OPTIONS 메소드의 응답 예제

(2) MKCOL 메소드

MKCOL 메소드는 WebDAV 서버에 새로운 컬렉션을 생성하는 메소드로 DavUP을 지원하는 WebDAV 서버에 새로운 컬렉션을 생성할 때 (그림 7)과 같이 DavUP-Type 헤더에 설정하려는 컬렉션 속성 타입에 대한 URI를 같이 포함하여 요청한다. 이러한 요청에 대하여 DavUP을 지원하는 WebDAV 서버는 해당 URI가 가리키는 자원에 설정된

TypeDefinition 속성을 확인한다. 그리고 TypeDefinition 속성에서 정의하고 있는 사용자 속성을 해당 자원에 설정한다.

```
MKCOL /webdisc/xfiles/ HTTP/1.1
DavUP-Type: /App/Col/ctype1
```

(그림 7) MKCOL 메소드의 요청 예제

정상적으로 요청이 수행되었을 때는 상태번호 201 응답 메시지를 돌려준다. 만약 해당 DavUP-Type에 정의된 사용자 속성 타입 URI가 존재하지 않거나 WebDAV 서버가 DavUP을 지원하지 않을 경우에는 DavUP-Type 헤더를 무시하게 된다.

(3) PUT 메소드

PUT 메소드는 WebDAV 서버에서 새로운 자원을 등록하기 위한 메소드이다. WebDAV 서버에 새로운 자원을 등록할 때 DavUP-Type 헤더에서 지정한 문서 타입에서 정의하고 있는 사용자 속성을 해당 자원이 정상적으로 등록된 후에 자동으로 설정한다. (그림 8)은 PUT 메소드를 통해서 hello.html를 WebDAV 서버에 등록하면서 myfiletype에서 정의한 사용자 속성을 설정하는 예를 보여주고 있다.

```
PUT /webdisc/xfiles/hello.html HTTP/1.1
DavUP-Type: /App/Doc/myfiletype
```

(그림 8) PUT 메소드의 요청 예제

4. DavUP을 지원하는 WebDAV 서버의 구현

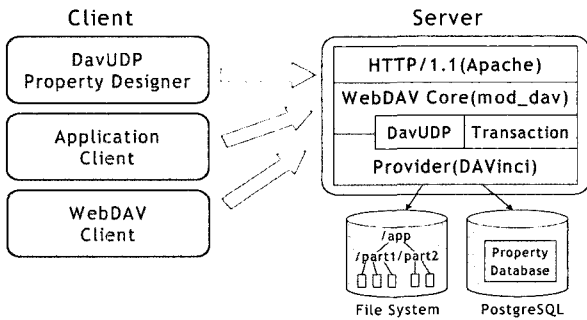
4.1 DAVinci 서버의 소개

본 논문에서 제안한 DavUP 프로토콜의 구현 실험을 위하여 DavUP을 지원하는 새로운 WebDAV 서버를 구현하였다. 구현된 서버는 다양한 WebDAV 서버들 중에서 공개 소프트웨어 기반의 아파치 mod_dav 모듈을 응용한 것이다. 아파치 mod_dav 모듈은 WebDAV 메소드를 처리하는 모듈과 자원에 대한 내용과 속성을 관리하기 위한 제공자(Provider) 모듈로 구분이 된다. 그런데 이 제공자 모듈이 파일 기반으로 되어 있어 속성을 저장하고 관리하는데 있어서 관리 능력이나 기능이 매우 제한적이다. 또한 MySQL 데이터베이스를 제공자 모듈로 제공하도록 구현한 Catacomb의 경우에는 WebDAV 자원에 대한 내용과 속성을 모두 데이터베이스의 필드로 저장하도록 함으로써 대용량의 자원을 관리하는데 있어 그 기능이 제한될 가능성이 있다.

본 연구에서는 이러한 파일 시스템 기반의 모듈과 순수 데이터베이스 기반의 모듈의 기능을 통합하여 WebDAV 자원의 내용은 파일 시스템 형태로 관리하고 속성 정보만을 PostgreSQL 데이터베이스로 관리할 수 있도록 새로운 제공자 모듈을 개발하고 이를 DAVinci(WebDAV Is New

Collaborative web-authoring Innovation)라고 명명하였다. 그리고 DAVinci WebDAV 서버를 기반으로 하여 DavUP 명세의 기능을 확장 구현하였다. DAVinci가 DavUP을 지원하기 위해서는 DAVinci의 mod_dav 모듈 단계의 확장과 제공자 모듈 단계의 확장으로 구분된다[15].

(그림 9)는 DavUP을 지원하는 DAVinci 서버의 전체적인 시스템 구조이다. 클라이언트는 기존의 WebDAV 클라이언트와 함께 응용 시스템을 설치하기 위한 DavUP 설계 도구와 구현된 WebDAV 기반의 응용 시스템 서버에 접속하기 위한 응용 시스템 클라이언트로 구분된다. DavUP을 지원하는 WebDAV 서버는 아파치 웹 서버와 WebDAV 메소드를 처리하는 mod_dav 모듈을 기반으로 PostgreSQL을 속성 저장소로 이용하는 DAVinci 제공자 모듈로 구성되어 있다. 그리고 그 내부에 DavUP을 지원하는 기능을 추가하였다.



(그림 9) DavUP을 지원하는 DAVinci 서버의 구조

DavUP 기능은 기존의 mod_dav 모듈에서 수행하는 HTTP 메소드의 수행 단계에서 본 논문에서 정의한 DavUP-Type 헤더를 인식하여 적절한 처리를 하는 기능을 수행한다. DavUP을 지원하기 위해서는 일부는 mod_dav 모듈을 확장하여야 하고, 일부는 DAVinci 제공자 모듈을 확장한다.

4.2 DAVinci 서버의 확장 구현

DavUP을 지원하기 위한 DAVinci 시스템에 대한 확장의 핵심은 DavUP에서 새롭게 정의한 DavUP-Type 헤더에 대한 처리와 확장된 WebDAV 메소드에 대한 처리이다. 따라서 DavUP을 지원하기 위하여 WebDAV의 메소드별로 적절한 모듈의 수정이 필요하다. OPTIONS 메소드의 경우, 아파치의 mod_dav 모듈을 직접 수정하여야 한다. 그리고 나머지 메소드들은 mod_dav 모듈에서 후크 함수(Hook Function)에 의해서 호출되는 제공자 모듈인 DAVinci 모듈을 수정함으로써 적용이 가능하다.

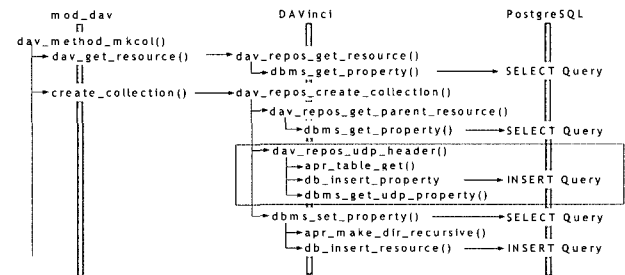
DAVinci 모듈의 확장은 새로운 WebDAV 자원의 생성을 요구하는 MKCOL과 PUT 메소드를 처리하는 부분에서 발생한다. DavUP 명세에서 기술한 바와 같이 DavUP-Type 헤더 정보를 해석하여 사용자 속성으로 저장하고, 헤더에서 지정하고 있는 속성 타입 URI를 통하여 해당 자원에 설정할 사용자 속성 타입을 해당 자원에 설정한다.

4.2.1 OPTIONS 메소드

OPTIONS 메소드는 웹 서버의 해당 자원에 대한 특성 정보를 요청하는 메소드이다. DavUP에서 정의한 DavUP-Type 헤더 정보를 클라이언트가 WebDAV 요청을 하면 그 응답으로 응답 몸체에 포함하여 리턴한다. OPTIONS 메소드에 대한 처리는 제공자 모듈 단계까지 처리를 요구하지 않기 때문에 기본적으로 mod_dav 모듈에서 전적으로 처리를 한다.

4.2.2 MKCOL 메소드

MKCOL 메소드는 새로운 WebDAV 컬렉션을 생성하는 메소드로 컬렉션을 생성하기 위해서는 기본적으로 WebDAV 서버의 파일 저장소에 새로운 컬렉션을 위한 디렉터리를 생성하고 속성 관리를 위한 데이터베이스 처리 작업을 수행한다. 따라서 요청의 헤더에서 지정하고 있는 DavUP-Type 값을 이용하여 해당 타입의 속성 정보를 데이터베이스에서 직접 확인하여 자원의 사용자 속성으로 설정한다.



(그림 10) MKCOL 메소드의 확장 부분

(그림 10)은 MKCOL 메소드의 확장 부분을 모듈의 전체 흐름과 함께 보여주고 있다. 기본적으로 DavUP-Type 헤더에서 지정하고 있는 타입 컬렉션의 속성 설정 내용은 PROPFIND를 통해서 조회하고 그 결과를 새롭게 생성되는 컬렉션에 사용자 속성으로 설정한다. 그러나 DAVinci 모듈에서는 내부적으로 속성을 데이터베이스에서 관리하고 있는 관계로 직접 데이터베이스의 정보를 읽어서 처리하였다.

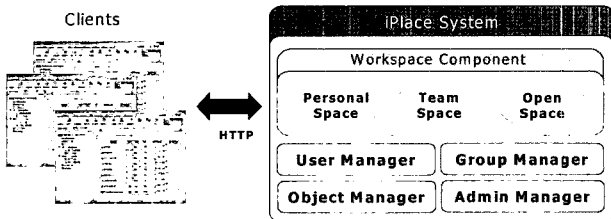
4.2.3 PUT 메소드

PUT 메소드는 WebDAV 컬렉션 내에 자원을 등록하는 메소드이다. PUT 메소드는 WebDAV 서버의 파일 저장소에 새로운 자원을 생성하고 속성 관리를 위한 데이터베이스 처리 작업을 수행한다. 따라서 기본적인 처리 절차가 MKCOL 메소드와 동일하다. mod_dav 모듈의 dav_method_put() 함수에서 호출을 받아 새로운 자원을 추가하는 부분에서 DavUP-Type 헤더 정보를 먼저 처리하도록 하였다.

5. 활용 사례

콘텐츠 기반의 응용 시스템을 개발할 때, 콘텐츠를 웹상에서 쉽게 접근하고 분산 저작도 지원하는 WebDAV 기반으로 개발한다면 응용 시스템 개발이 용이해 질 것이다. 콘텐츠 기

반의 응용 시스템은 문서관리 시스템, 환경관리 시스템, 협업 지원 시스템 등이 있다. 이러한 응용 시스템을 WebDAV 기반의 응용 시스템으로 개발할 수 있도록 DavUP 프로토콜을 제안하였다. 본 장에서는 DavUP 프로토콜의 유용성을 검증하기 위한 활용 사례로서 웹 기반의 협업 지원 시스템의 일부분인 공개작업장을 DavUP 프로토콜 기반으로 구현하고자 한다. 다양한 웹 기반의 협업 지원 시스템 중에서 본 연구실에서 개발한 iPlace[16] 협업시스템을 기초로 하였다. iPlace 시스템의 작업장은 (그림 11)과 같이 개인작업장, 팀작업장, 공개작업장으로 구성되어 있다. 개인작업장은 개인의 작업 공간 및 자료저장을 위한 목적으로 사용되며, 팀작업장은 부서 그룹이나 공동작업을 필요로 하는 구성원들 사이의 자료 공유를 위한 장소로 사용된다. 공개작업장은 다수의 일반 사용자를 대상으로 공개된 자료의 공유 및 교환 기능을 제공한다[16].



(그림 11) iPlace 협업시스템의 구성

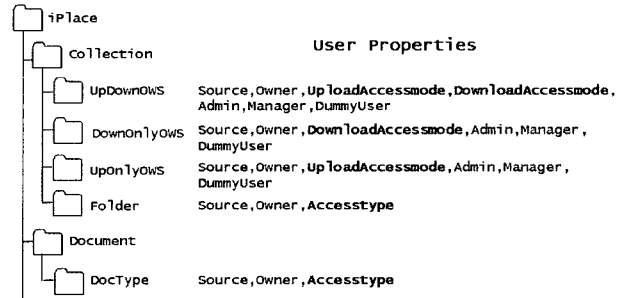
공개작업장은 웹 기반의 공동작업 공간으로서 공간 관리를 위해서 폴더의 개념을 이용하여 자료를 효과적이고 체계적으로 분류할 수 있도록 제공한다. 공개작업장은 작업장의 성격에 따라 세 가지 타입의 작업장 생성기능을 제공한다. 작업장을 접근하는 사용자들은 누구나 자료를 업로드(Upload)하고 다운로드(Download)할 수 있는 “업로드 및 다운로드 지원 작업장”, 공개작업장을 생성한 관리자가 웹을 통한 자료의 배포를 목적으로 하는 “다운로드 전용 작업장”, 그리고 업로드만을 허용하여 자료의 수집이나 제출을 목적으로 하는 “업로드 전용 작업장”으로 분류된다. 기존의 공개작업장은 관계형 데이터베이스를 이용하여 작업장의 자원을 등록하고 관리하도록 하였다[17]. 데이터베이스를 이용한 관리는 자원과 관련 속성을 별도의 테이블 형태로 관리하도록 구현하여야 한다. 또한 특정 데이터베이스에 한정되어 개발될 수밖에 없다. WebDAV를 이용한 공개작업장의 설계와 구현은 자원에 대한 속성을 XML 형태의 값으로 설정함으로써 쉽게 적용할 수 있는 장점이 있다.

공개작업장의 성격과 공개작업장을 구성하는 데이터 모델을 분석하여 공개작업장의 자원 관리를 WebDAV 기반으로 구성하였다. 먼저 WebDAV 기반으로 공개작업장을 적용하기 위해서는 공개작업장의 특성과 데이터 모델을 WebDAV 자원으로 매핑하는 작업이 필요하다. 이러한 과정에서 WebDAV 자원에 설정되어야 하는 다양한 종류의 사용자 속성들이 나타난다. 분류된 사용자 속성은 WebDAV 자원 저장소에 자원에 등록될 때 적절하게 분류된 사용자 속성을 설정

하여야 한다.

iPlace 협업시스템의 공개작업장을 DavUP 프로토콜에서 정의하고 있는 사용자 속성 정의의 형태로 설계하면 (그림 12)와 같은 컬렉션 구조와 속성 타입으로 설계할 수 있다. 공개작업장에 적용되는 형 정의의 컬렉션을 UpDownOWS, DownOnlyOWS, UpOnlyOWS 타입으로 분류하고 각 타입에 맞는 사용자 속성을 정의하였다. 물론 각각의 타입이 동일한 사용자 속성을 가진다. 하지만 각각의 타입 자체가 다른 네임스페이스를 가지기 때문에 응용 시스템에서 적절하게 구분하여 사용할 수도 있다. 이러한 설계를 바탕으로 공개작업장을 위한 사용자 속성의 설계 결과를 WebDAV 서버에 적용을 시키고 iPlace 협업시스템을 위한 클라이언트를 개발하였다. 실제로 클라이언트에서 공개작업장을 생성할 때 MKCOL 메소드를 (그림 13)과 같이 요청하면 해당 공개작업장 컬렉션을 생성하면서 업로드/다운로드를 지원하는 공개작업장으로 설정된다. 클라이언트에서 새폴더 생성 아이콘을 클릭하고, 생성하는 공개작업장의 이름과 작업장의 타입을 정하고 생성 버튼을 클릭하면, HTTP 요청 정보 화면의 요청이 서버로 전송이 되어 새로운 공개작업장이 생성된다.

Configuration Collection



(그림 12) 공개작업장을 위한 DavUP 환경 설정 컬렉션 구조

```

MKCOL /CoWorkSpace/OpenWorkspace/Papers/HTTP/1.1
Host: iplace.ulsan.ac.kr
DavUP-Type: /iPlace/Collection/UpDownOWS
    
```

(그림 13) 공개작업장을 생성하는 MKCOL 메소드의 요청 예제

6. 결 론

본 논문에서는 WebDAV 사용자 속성 관리의 효율적 지원을 위한 DavUP 프로토콜을 정의하였다. DavUP 프로토콜은 WebDAV의 사용자 속성을 이용하여 다양한 응용 시스템의 개발을 체계적으로 지원한다. DavUP에서는 응용 시스템의 자원에 적용될 사용자 속성 타입을 관리하는 응용 시스템을 위한 컬렉션 구조를 정의하고 있다. 컬렉션 구조는 WebDAV 서버에 응용 시스템을 위한 컬렉션과 문서의 사용자 속성을 집합화한 타입 정보를 가진다. 다양한 사용자 속성을 정의한 형 정의의 속성은 형의 이름으로 생성된 컬렉션의 새로운 사용자 속성인 TypeDefinition 속성이다.

WebDAV 서버가 DavUP을 지원하기 위해서는 새로운 헤더의 추가와 기존 메소드의 확장이 필요하다. *DavUP-Type* 헤더는 WebDAV 서버가 DavUP을 지원하는지를 확인하고 새로운 컬렉션이나 문서를 생성할 때 설정할 타입 정보의 URL 값을 가진다. WebDAV 메소드의 확장은 이러한 DavUP-Type 헤더를 인식하고 형 정의 속성에서 정의한 사용자 속성을 해당 자원에 설정하는 역할을 수행한다. 이와 같이 DavUP 프로토콜을 지원할 수 있도록 자체 개발한 *DAVinci* WebDAV 서버를 확장하였으며 클라이언트에 의해서 관리되어야 할 사용자 속성들의 관리 기능이 서버에서 이루어질 수 있도록 하였다. 확장된 *DAVinci* 서버를 이용하여 사용자에게 협업을 지원하는 공간인 일반적인 공개작업장을 실험적으로 개발하여 프로토콜의 유용성을 검증하였다.

향후 과제로는 응용 시스템 컬렉션 구조에 대한 다양한 계층의 사용자들의 접근 문제를 해결하기 위한 방안으로 WebDAV의 확장 프로토콜인 접근 제어 프로토콜과의 연계에 대한 연구를 수행할 계획이다. 또한 응용 시스템 기반 설계를 위한 사용자 속성 설계 도구와 DavUP을 이용한 WebDAV 기반의 다양한 응용 시스템을 개발할 예정이다.

참 고 문 헌

- [1] E. James Whitehead Jr. and Meredith Wiggins, "WEBDAV: IETF Standard for Collaborative Authoring on the Web," IEEE Internet Computing, pp.34~40, September/October, 1998.
- [2] Y. Goland, E. Whitehead, A. Faizi, S. Carter and D. Jensen, "HTTP Extensions for Distributed Authoring-WEBDAV," RFC 2518, Standards Track, February, 1999.
- [3] E. James Whitehead Jr. and Yaron Y. Goland, "The WebDAV property Design," Software: Practice and Experience, Vol. 34 issue 2, pp.135~161, February, 2004.
- [4] G. Clemm, J. Amsden, T. Ellison, C. Kaler and J. Whitehead, "Versioning Extensions to WebDAV(Web Distributed Authoring and Versioning)," RFC 3253, 2002.
- [5] G. Clemm, J. Reschke, E. Sedlar and J. Whitehead, "Web Distributed Authoring and Versioning(WebDAV) Access Control Protocol," RFC 3744, May, 2004.
- [6] S. Reddy, J. Davis, A. Babich and J. Reschke (Ed.), "WebDAV Search: draft-reschke-webdav-search-06," <http://greenbytes.de/tech/webdav/draft-reschke-webdav-search-latest.html>, IETF, 2003.
- [7] 정혜영, 안건태, 유양우, 박양수, 이명준, "DavSUDP: 웹데브 사용자 정의 속성의 활성화를 위한 단순 프로토콜", 정보처리학회논문지C, 제12-C권 제1호, pp.129~136, 2005.
- [8] 정혜영, 안건태, 박양수, 이명준, "웹데브를 지원하는 닷넷 기반의 탐색기 개발", 정보처리학회논문지C, 제11-C권 제5호, pp.1~8, 2004.
- [9] Greg Stein, "mod_dav: a DAV module for Apache," http://www.webdav.org/mod_dav/, November, 5, 2001.
- [10] Sung Kim, Kai Pan and Elias Sinderson, "mod_dav_dbms: A Database Backed DASL Module for Apache," Department of Computer Science, University of California at Santa Cruz, March, 14, 2002.
- [11] Karen Schuchardt, James Myers and Eric Stephan, "A Web-based Data Architecture for Problem Solving Environments: Application of Distributed Authoring and Versioning to the Extensible Computational Chemistry Environment," Pacific Northwest National Laboratory, Cluster Computing 5(3), pp.287~296, 2002.
- [12] Sunghun Kim, Mark Slater and E. James Whitehead, Jr., "WebDAV-based Hypertext Annotation and Trail System," Hypertext 2004, Santa Cruz, California, USA, August, 9~13, 2004.
- [13] Joachim Feise, "Posties: A WebDAV Application for Collaborative Work," Hypertext 2000, San Antonio, TX., May, 30~June, 3, 2000.
- [14] Lisa Dusseault, 'WebDAV: Next Generation Collaborative Web Authoring', Prentice Hall, 2003.
- [15] 정혜영, 안건태, 박양수, 이명준, "아파치 기반의 신뢰성 있는 자원관리를 지원하는 웹데브 서버", 정보처리학회논문지C, 제11-C권 제4호, pp.545~554, 2004.
- [16] 안건태, 정명희, 이근웅, 문남두, 이명준, "iPlace: EJB 기술을 이용한 웹 기반 협업시스템", 정보처리학회논문지D, 제8-D권 제6호, pp.735~746, 2001.
- [17] 박희종, 김진홍, 신원준, 박양수, 이명준, "iPlace 시스템에서의 공개 작업공간 지원", 한국정보과학회, 봄학술발표논문집(A), 제31권 제1호, pp.478~480, 2004.



정 혜 영

e-mail : hyjung@utp.or.kr
 1994년 울산대학교 전자계산학과(공학사)
 1998년 울산대학교 대학원 전자계산학과(공학석사)
 2005년 울산대학교 대학원 컴퓨터정보통신공학과(공학박사)

1993년~1995년 (주)현대중공업 조선사업부 근무
 1997년~2005년 (주)LG CNS 기술서비스부문 근무
 2003년~2004년 울산대학교 컴퓨터정보통신공학부 객원교수
 2005년~현재 (재)울산산업진흥테크노파크 정밀화학지원센터 선임연구원
 관심분야: WebDAV, 협업지원시스템, 분산 시스템



김 동 호

e-mail : herokim11@mail.ulsan.ac.kr
2005년 울산대학교 컴퓨터정보통신공학부
(공학사)
2005년 현재 울산대학교 컴퓨터정보통신
공학부 석사과정
관심분야: 협업시스템, 웹 프로그래밍, 분
산 객체 시스템



이 명 준

e-mail : mjlee@ulsan.ac.kr
1980년 서울대학교 수학과(학사)
1982년 한국과학기술원 전산학과(석사)
1991년 한국과학기술원 전산학과(박사)
1982년~현재 울산대학교 컴퓨터정보통신
공학부(교수)
1993년~1994년 미국 버지니아대학 교환교수
관심분야: 웹기반 정보시스템, 프로그래밍언어, 분산 프로그래
밍, 생물정보학



안 건 태

e-mail : seeahn@gmail.com
1999년 울산대학교 전자계산학과(공학사)
2001년 울산대학교 대학원 컴퓨터정보통
신 공학과(공학석사)
2005년 울산대학교 대학원 컴퓨터정보통
신 공학과(공학박사)

관심분야: 분산시스템, 협업시스템, WebDAV, 웹기반기술