

아파치 기반의 신뢰성 있는 자원관리를 지원하는 웹데브 서버

정혜영* · 안건태** · 박양수* · 이명준†

요약

WebDAV(Web-based Distributed Authoring and Versioning, RFC 2518)는 인터넷을 통하여 원거리에 위치한 다수의 작업자들 간의 협업을 지원하기 위한 프로토콜이다. 웹데브(WebDAV)는 웹 통신 프로토콜인 HTTP/1.1의 확장으로 인터넷을 통하여 다양한 콘텐츠의 비동기적인 협업을 지원하기 위한 프로토콜을 제공한다. 웹 기반의 협업지원 시스템이나 문서관리 시스템과 같은 기존의 응용 시스템에서 웹데브 기능을 제공하기 위해서는 이러한 시스템들이 웹데브의 메서드(method)와 헤더(header) 정보를 다루는 추가적인 구현이 필요하다. 본 논문에서는 웹데브 명세를 지원하는 아파치 기반의 *DAVinci(WebDAV Is New Collaborative web-authoring Innovation)* 웹데브 서버를 개발하였다. 공개 소프트웨어 기반인 아파치(Apache) 웹 서버에는 웹데브를 지원하기 위한 `mod_dav` 모듈을 제공하고 있다. DAVinci는 `mod_dav` 모듈의 서비스 제공자 형태로 추가되어 웹데브의 자원과 속성 관리를 담당한다. 본 시스템은 자원을 파일 시스템에 저장하고 속성 정보를 PostgreSQL 데이터베이스로 관리하며 추가적으로 자원과 속성들 간에 비일관성이 발생하지 않도록 보장하는 일관성 관리자(consistency manager)를 제공한다.

An Apache-based WebDAV Server Supporting Reliable Resource Management

Hye-Young Jung* · Geon-Tae Ahn** · Yang-Soo Park* · Myung-Joon Lee†

ABSTRACT

WebDAV is a protocol to support collaboration among the workers in geographically distant locations through the Internet. WebDAV extends the web communication protocol HTTP/1.1 to provide a standard infrastructure for supporting asynchronous collaboration for various contents across the Internet. To provide the WebDAV functionality in legacy applications such as web-based collaborative systems or document management systems, those systems need to be implemented additionally to handle the WebDAV methods and headers information. In this paper, we developed an Apache-based WebDAV server, named *DAVinci(WebDAV Is New Collaborative web-authoring Innovation)* which supports the WebDAV specification. DAVinci was implemented as a form of service provider on a `mod_dav` Apache module. `Mod_dav`, which is an Apache module, is an open source module to provide WebDAV capabilities in an Apache web server. We used a file system for storing resources and the PostgreSQL database for their properties. In addition, the system provides a consistency manager to guarantee that both resources and properties are maintained without inconsistency between resources and their properties.

키워드 : 웹데브(WebDAV), HTTP, `mod_dav`, DAVinci, 일관성 관리자(Consistency Manager)

1. 서론

인터넷의 발달로 인하여 지역적으로 원거리에 위치한 다수의 작업자 그룹들이 공동 작업을 수행할 수 있는 다양한 기술들이 개발되었다. 초기 공급업체들은 협업을 지원하기 위하여 HTTP(Hyper-Text Transfer Protocol) 프로토콜을 활용하거나 또는 각자의 고유 프로토콜들을 HTTP 프로토

콜에 추가 정의하는 형태로 구현하였다. 웹 기반의 협업 지원 시스템으로는 *iPlace*[1], *BSCW*[2] 등이 있다. HTTP 프로토콜에 각자의 고유 프로토콜의 추가를 통한 협업 지원은 공급업체들 간의 상호 운용성이 결여되는 문제가 발생하였다[3, 4]. 공급업체들 간에 상호 운용성을 보장하기 위한 웹 기반의 분산 저작과 버전관리 표준의 필요성이 증가되었고, 이러한 요구를 충족시키기 위하여 W3C(World Wide Web Consortium)의 IETF(Internet Engineering Task Force) 산하의 작업 그룹에 의해서 웹데브(WebDAV) 명세가 1999년 2월에 발표되었다[5].

* 본 논문은 2003년 한국소프트웨어진흥원의 “공개 S/W 프로젝트 활용 S/W 전문 인력 양성 사업” 지원으로 이루어 졌음.

† 정희원 : 울산대학교 컴퓨터정보통신공학부 교수

** 준희원 : 울산대학교 대학원 컴퓨터정보통신공학부
논문접수 : 2004년 5월 21일, 심사완료 : 2004년 7월 5일

웹데브는 HTTP/1.1[6]을 확장한 웹 통신 프로토콜로서 인터넷을 통하여 다양한 콘텐츠의 비동기적인 협업 제작을 지원하기 위한 표준 하부구조를 제공한다[5]. 웹데브 명세를 지원하는 다양한 종류의 서버들은 상호간의 협업 작업이 가능하게 되었다. 그리고 클라이언트 응용 프로그램이 웹데브를 지원함으로써 이기종의 웹데브 서버들을 통하여 분산 저작을 수행할 수 있게 되었다. 웹데브를 지원하는 대표적인 서버 제품으로는 마이크로소프트사의 IIS 5.0[7], 아파치(Apache)의 mod_dav[8] 모듈, UCSC의 Catacomb[15], Software AG사의 Tamino Server[9] 등이 있으며, 클라이언트 제품으로는 마이크로소프트사의 MS-Office 2000[10], SouthRiver Technology사의 WebDrive[11], Xythos사의 WebFile Client[12], 본 연구실의 닷넷기반 웹데브 탐색기[13] 등이 있다.

아파치 mod_dav 모듈은 아파치 웹 서버의 모듈 형태로 웹데브 서버의 기능을 구현하고 있다. 웹데브 명세에 정의된 확장 프로토콜의 처리 권한은 아파치 서버의 환경 설정을 통하여 아파치 웹 서버에서 mod_dav 모듈로 처리 권한이 넘겨진다. mod_dav 모듈은 웹데브 프로토콜의 처리와 적절한 저장소 관리를 통하여 자원과 속성의 관리를 처리한다. mod_dav 모듈에는 기본적으로 하위 모듈로 mod_dav_fs라는 파일 시스템 기반의 자원과 속성의 관리를 처리하는 저장소 관리 모듈을 제공한다. 이 모듈은 개별 웹데브 자원의 속성 정보를 파일 형태로 관리하기 때문에 데이터 크기의 제한이나 검색의 문제점 등을 안고 있다. 이러한 한계를 해결하기 위하여 본 논문에서는 자원만을 파일시스템 형태로 관리하고 속성 정보는 데이터베이스로 관리하는 모델을 개발하였다.

본 논문에서는 새로운 아파치의 모듈을 설계하여 mod_dav 모듈의 하위 모듈로 등록하도록 구현하였다. 데이터베이스로는 공개 소프트웨어 기반의 PostgreSQL 관계형 데이터베이스를 사용하고 있다. PostgreSQL은 기본적으로 트랜잭션(transaction)을 제공하고 있어 속성 정보의 일관성 유지를 지원하는데 적합하다고 판단된다. 웹데브 자원의 속성 정보를 데이터베이스로 관리함으로써 안정적인 속성 관리가 가능하게 되었다. 그러나 파일과 데이터베이스로 자원과 속성을 분리하여 관리하기 때문에 자원과 속성 정보 간의 일관성의 문제가 발생할 수 있다. 따라서 본 논문에서는 신뢰성 있는 자원 관리를 지원하고 일관성의 문제를 해결하기 위하여 일관성 관리자(consistency manager)를 별도로 두도록 설계하였다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서 관련 연구의 소개를 통하여 웹데브에 대한 소개와 아파치 모듈인 mod_dav와 Catacomb 시스템에 대하여 살펴보고 3장

에서는 웹데브 서버의 설계와 관련하여 기존 웹데브 서버의 분석과 구현 시스템의 설계, 신뢰성 있는 자원관리 방안 등에 대하여 기술한다. 4장에서는 설계된 모델에 대한 구현에 대하여 기술하고 5장에서 실험결과와 기능비교를 하고 끝으로 6장에서 결론과 향후 연구 방향에 대하여 기술한다.

2. 관련 연구

본 장에서는 웹데브 프로토콜에 대한 기본적인 소개와 아파치 웹 서버에서 웹데브를 지원하도록 구현된 mod_dav 모듈에 대하여 기술한다. 또한 데이터베이스 형태로 자원과 메타데이터를 함께 저장하도록 아파치 모듈로 구현된 Catacomb[15]의 기본 구성에 대하여 기술한다.

2.1 웹데브(WebDAV)

웹데브는 인터넷을 통하여 광범위하고 다양한 콘텐츠의 비동기적인 협업 저작을 지원하기 위한 프로토콜이다. 웹데브는 HTTP/1.1 프로토콜의 확장을 통하여 사용자에게 원격 서버들의 파일들을 수정하고 관리할 수 있도록 한다. 웹데브 표준 명세는 W3C의 IETF 웹데브 작업 그룹에 의해서 1999년 2월에 발표되었다. 웹데브의 주요 기능으로는 잠금 관리(Lock Management), 속성 관리(Property Management), 컬렉션(Collection), 이름 공간 관리(Namespace management) 등이 있으며 이들에 대한 개략적인 설명은 다음과 같다[5].

2.1.1 잠금 관리

웹데브에서는 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 저장시킴으로 인하여 변경 내용을 잊어버리는 갱신 분실 문제를 방지한다. 웹데브는 잠금 소유자만이 잠금된 자원에 대한 갱신을 허용하는 배타적 쓰기 잠금(exclusive write lock)과 자원에 대한 협업 작업자 그룹에게 공동 작업을 허용하는 공유된 쓰기 잠금(shared write lock)을 제공한다. 웹데브 잠금은 TCP 연결과 독립적으로 존재하기 때문에 잠금 토큰을 가지고 네트워크로부터 연결이 끊어진 상태로 작업을 하고난 후, 다시 연결하여 수정내용을 갱신할 수 있다.

2.1.2 속성 관리

웹데브 속성 관리는 웹 자원에 관한 저자, 마지막 수정일자 등과 같은 정보에 대한 생성, 삭제 그리고 검색을 할 수 있도록 한다. 웹데브 속성은 (이름, 값)의 쌍으로 이루어져 있다. 속성 이름은 URL이며, 속성 값은 잘 정의된 XML

(eXtensible Markup Language) 문서이기 때문에 광범위한 XML 형태의 데이터 저장이 가능하다.

2.1.3 컬렉션

웹데브는 자원을 저장할 수 있는 컬렉션 개념을 도입하고 있다. 웹데브를 통한 자원 관리에는 컬렉션 자체와 컬렉션 내에 있는 자원이나 파일에 대한 생성, 이동, 복사 그리고 삭제 기능을 포함한다. 웹데브 컬렉션은 컬렉션 내의 자원에 대한 직접 포함과 웹 상의 어떤 곳에 위치한 자원에 대한 참조적인 포함을 제공한다. 직접 포함 관계의 일관성은 웹데브 서버에 의해 관리된다.

2.1.4 이름 공간 관리

이름 공간 관리는 서버의 이름 공간 내에 있는 웹 문서를 복사하고 이동할 수 있는 능력과 자원을 위한 저장 공간인 컬렉션의 콘텐츠를 생성하고 내용을 보여준다. HTTP는 개별 자원에 대한 직접 접근만을 처리하지만 웹데브는 데이터를 보다 효율적으로 구성하는 수단을 제공한다.

2.2 웹데브 관련 아파치 모듈

아파치는 웹 서버 시장에서 2004년 3월 기준으로 70% 이상의 점유율을 차지하고 있을 정도로 인터넷에서 가장 널리 사용되는 웹 서버 소프트웨어이다[14]. 아파치 웹 서버의 구조는 클라이언트의 요청을 처리하기 위한 다양한 모듈들로 구성되어 있다. 사용자는 서버의 환경 설정을 어떻게 하느냐에 따라 다양한 방법으로 작동된다. 이러한 결과로 아파치는 매우 유연하고 확장 가능한 형태를 가진다. 즉, 모듈화된 접근을 사용하여 아파치의 주 소스 코드를 변경하지 않고 서버 측의 기능을 구현하는 것이 가능하다. 이것은 아파치 구조의 효율성과 견고성의 영향으로 개발자는 서버의 소스 코드를 재 구현할 필요가 없다. 아파치는 요청을 처리하기 위하여 초기화, 메서드 가져오기, 요청 읽기 등의 여러 가지 구분된 단계를 가진다. 서버 측의 각 단계별 작업 수행 내에 추가적인 처리 단계를 끼워 넣는 것이 가능한데 이러한 추가 기능을 모듈 형태로 구현한다[15, 16].

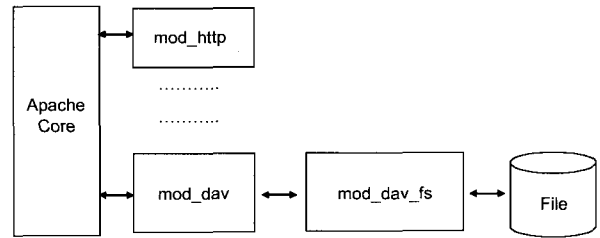
2.2.1 아파치 mod_dav

웹데브 프로토콜을 처리하는 모듈은 아파치 서버 내에서 (그림 1)과 같이 2개의 모듈로 나뉜다. 주 모듈인 mod_dav는 메서드 요청들의 처리를 다루며 mod_dav_fs 모듈은 자원의 저장소 기능을 다루기 위한 인터페이스를 처리한다. mod_dav_fs에서는 저장소를 표준 파일 시스템 기반으로 구현하여 특정 디렉터리 내에 자원별로 메타데이터를 파일로 저장하고 있다.

mod_dav_fs는 파일 시스템의 파일과 디렉터를 이용하여

데이터 객체와 컬렉션들의 영속성을 제공하고 있다. 메타데이터는 각각의 문서나 컬렉션 단위로 하나의 파일 형태로 데이터베이스 관리자(DBM)내의 해쉬 테이블로 저장된다. 사용되는 DBM은 파일 시스템 기반의 데이터베이스인 Simple DBM(SDBM)이나 Gnu DBM(GDBM)이 있다. SDBM은 1KB 크기로 개별 메타데이터의 값을 제한하고 있고 초기 크기로 8KB를 가진다. GDBM은 메타데이터 저장 공간의 크기의 제한은 없으며 좀 더 나은 성능을 가지고 있고 초기 데이터베이스 크기를 25KB를 가진다. 모든 DBM은 수작업에 의한 가비지 컬렉션 유틸리티를 통하여 변경되고 삭제된 메타데이터에 대한 공간을 정리해주어야 한다[17].

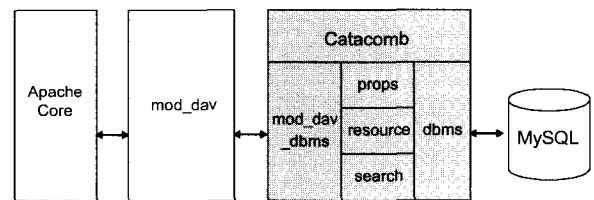
이것은 매우 직접적이고 쉬운 방법이지만 파일 시스템 구조 하에서 속성에 대한 검색 질의의 구현이 어렵고 기능의 구현에 있어서 많은 파일 열기와 닫기 처리를 야기한다. 따라서 웹데브 서버를 구현하는데 있어 대용량의 사용자 정의 속성의 사용과 다양한 검색 기능을 필요로 하는 경우에는 mod_dav_fs 모듈을 사용하는 것은 부적절하다.



(그림 1) 아파치 mod_dav 모듈의 구조

2.2.2 Catacomb

Catacomb[15]는 mod_dav_fs 모듈의 속성 저장 공간의 제한을 해결하고 보다 효율적인 검색을 지원할 수 있도록 데이터베이스를 기반으로 한 저장소 모델을 지원하는 시스템이다. (그림 2)는 Catacomb의 구조를 보여주고 있다. Catacomb는 공개 소스 기반의 웹데브 서버이며 mod_dav 모듈의 서비스 제공자 모듈을 MySQL 데이터베이스를 이용하여 구현하였다.



(그림 2) Catacomb의 구조

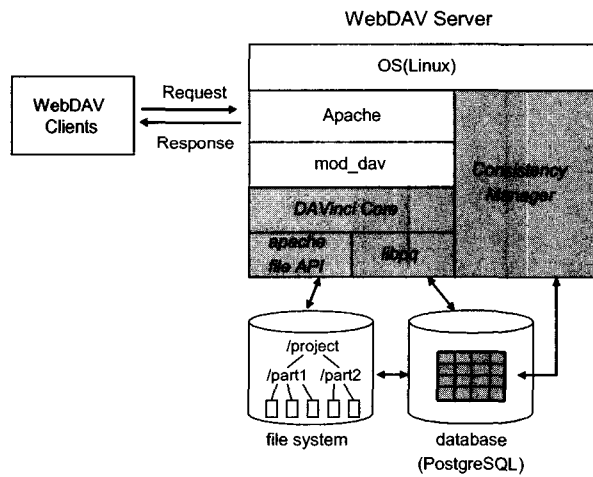
Catacomb에서는 웹데브의 자원과 속성을 모두 데이터베이스로 저장한다. 이렇게 함으로써 속성 검색과 자원 문서

내 검색 등의 기능 구현이 용이하다. 그러나 데이터베이스의 필드로 자원과 속성을 처리하기 위해서는 파일로 자원을 저장한 후에 파일을 데이터베이스에 저장하는 단계를 거쳐야 한다. 이러한 단계는 대용량의 자원을 저장하고 관리하는 웹데브 서버인 경우에 급격한 성능의 저하를 야기시킬 수 있다.

3. 시스템 설계

웹데브의 자원과 속성의 관리를 위해 본 논문에서는 자원을 파일 시스템 형태로 관리하고 자원의 메타데이터에 해당하는 속성을 데이터베이스로 관리하도록 서버 시스템을 설계하였다. 파일 시스템 형태로의 자원 관리는 웹데브의 이름 공간 관리를 파일과 디렉터리의 구조로 표현하고 있다. 그리고 웹데브 자원에 대한 속성 정보는 데이터베이스의 테이블로 설계하였다. 속성 정보는 속성 이름과 값으로 구성되어 있고 XML 형태로 정보 교환이 이루어지기 때문에 데이터베이스 형태로 표현되고 관리하는 것이 처리 성능이나 안정성이 더 뛰어나다. 또한 웹데브 속성을 데이터베이스 형태로 관리하는 것은 향후 확장된 웹데브 명세인 버전관리와 검색 등을 구현하는 것이 용이하다는 장점을 가진다.

(그림 3)은 설계한 웹데브 서버의 전체 구조를 나타낸다. 아파치 웹 서버와 아파치에서 웹데브를 지원하는 mod_dav 모듈을 통하여 웹데브 기능을 제공하고 있다. mod_dav 모듈은 웹데브 프로토콜을 해석하여 각 프로토콜에 해당하는 정의된 후크 함수(hook function)를 호출한다[16]. 하위 모듈인 서비스 제공자 모듈에서 이러한 후크 함수를 구현함으로써 웹데브 자원과 속성을 파일과 데이터베이스로 분리하여 관리하는 기능의 구현이 가능하다.



(그림 3) 전체 서버 시스템 구성도

DAVinci 서버의 구현을 위해 사용한 데이터베이스는 공개 소프트웨어 기반의 관계형 데이터베이스인 PostgreSQL을 이용하였다. PostgreSQL은 기본적으로 트랜잭션 처리 기능을 제공하기 때문에 웹데브 자원의 속성을 관리하는데 있어서 신뢰성 있는 일관성 유지가 가능하다. 물론 본 논문에서 제시한 모델은 자원과 속성을 분리하여 관리하기 때문에 자원과 속성 간의 일관성 유지에 대한 문제점을 가지고 있다. 예를 들어, 웹데브 자원은 파일 시스템에 대하여 정상적으로 생성, 변경, 삭제와 같은 기능이 처리되었지만 시스템의 장애로 인하여 해당 자원에 대한 속성 정보의 갱신 처리가 안 되는 경우나 그 반대의 경우가 발생하여 일관성을 유지하지 못하는 경우가 있을 수 있다. 이러한 문제점을 해결하기 위하여 (그림 3)에서와 같이 일관성 관리자를 제공한다.

일관성 관리자는 데이터베이스에 저장된 로그 정보를 통하여 파일정보와 해당 파일에 대한 속성 정보간의 불일치를 감시하고 해결하는 역할을 수행한다. 기본적으로 일관성 관리자는 웹데브 속성 정보를 기준으로 하여 파일의 불일치를 해결하도록 한다. 이것은 속성을 관리하는 데이터베이스가 트랜잭션을 좀 더 안정적으로 처리하기 때문이다. 자원의 속성 정보에 대하여 생성이나 변경이 정상적인 트랜잭션 처리로 이루어 졌을 때 그 트랜잭션 처리 내에서 로그 정보를 저장하고 있다. 이 로그 정보를 일관성 관리자에서 감시하고 해당 로그 정보에 대한 파일의 불일치 문제를 해결하는 형태이다.

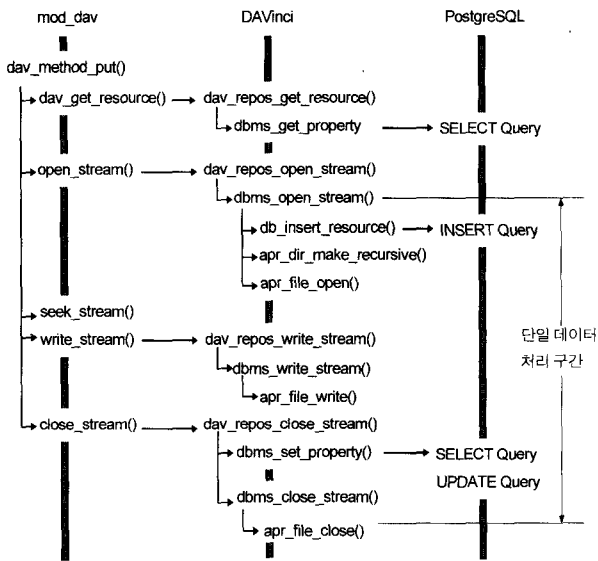
자원을 파일로 관리할 경우에는 웹데브가 아닌 방식으로 파일에 접근할 수 있다. 다른 방식에 의한 파일 접근으로 인하여 발생할 수 있는 파일에 대한 일관성과 신뢰성의 문제는 자원을 파일로 관리하는 모든 웹데브 서버에서 야기될 수 있는 문제이다. 이러한 파일에 대한 보안과 접근 권한은 운영체제에 의존한다. 따라서 파일 형태의 자원에 대한 신뢰성은 운영체제의 파일에 대한 보안 정책과 밀접한 관련을 가진다.

3.1 웹데브 메서드의 설계

신뢰성 있는 자원 관리를 지원할 수 있도록 웹데브 메서드를 처리하는데 있어 자원과 속성간의 일관성의 문제가 발생할 수 있는 부분에 대하여 분석하고 해결할 수 있도록 해야 한다. 웹데브 메서드 중에서 자원과 속성 정보의 변경이 발생할 수 있는 주요 메서드는 다음과 같다.

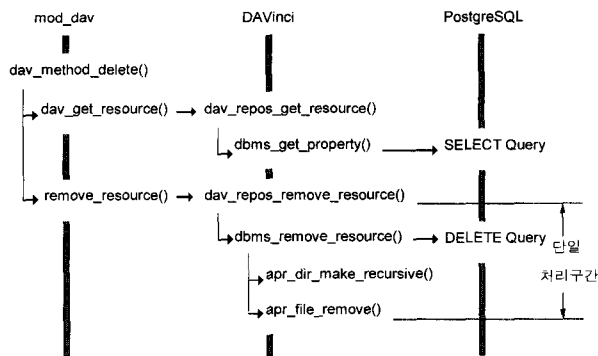
- **PUT Method** : 웹데브 서버에 새로운 자원을 추가하는 것으로 자원의 추가와 함께 해당 자원에 대한 기본적인 정보와 웹데브 속성 정보를 추가하게 된다. 이때 (그림

4)에서 보는 바와 같이 새로운 자원을 추가하기 위하여 먼저 데이터베이스에 속성 정보를 위한 데이터 열을 추가한다. 그리고 자원을 저장하기 위한 서버 상의 파일 시스템에 파일을 오픈하고 쓰기를 완료한다. 끝으로 자원 정보에 대한 데이터 열의 갱신과 파일 닫기를 수행하게 된다. 수행과정에서 데이터베이스 내에 데이터 열의 추가와 파일 등록 간에는 일관성을 유지하도록 구현하여야 한다.



(그림 4) PUT Method의 수행 절차

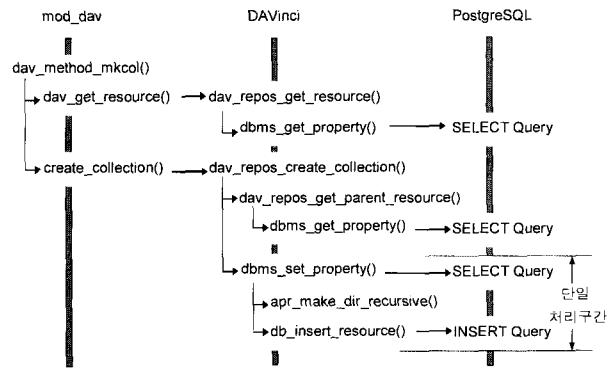
- DELETE Method** : 이 메서드는 웹데브 서버의 자원을 삭제하는 명령어로 (그림 5)와 같은 시나리오로 수행된다. (그림 5)에서 보는 바와 같이 데이터 열의 삭제와 파일의 삭제에 대하여 일관성을 유지할 수 있도록 처리하여야 한다.



(그림 5) DELETE Method의 수행 절차

- MKCOL Method** : 웹데브의 컬렉션을 생성하는 명령어로 이름 공간의 구조에 따라 컬렉션을 파일 시스템의 디

렉터리로 생성하게 되고 이 컬렉션에 대한 속성 정보를 (그림 6)에서와 같은 시나리오를 통하여 데이터베이스에 데이터 열을 추가한다. 디렉터리의 생성과 데이터 열의 추가에 대하여 일관성을 유지할 수 있도록 트랜잭션 영역 내에서 수행되도록 해야 한다.



(그림 6) MKCOL Method의 수행 절차

3.2 트랜잭션 모델의 설계

웹데브 서버에 있어서 트랜잭션은 협의의 의미로는 단일 메서드에 대하여 자원과 속성 정보의 생성과 변경에 대하여 원자성(atomicity)을 보장해야 한다는 의미를 뜻하고 광의의 의미로는 단위 작업에 대한 원자성을 보장해야 한다는 의미를 가진다. 예를 들어 웹 문서의 저장의 경우, 하나의 웹 페이지와 연결된 그림 파일 등이 저작(authoring) 작업에 있어 하나의 단위에 해당함으로써 등록을 할 경우 동시에 등록이 되거나 그렇지 않을 경우 모두 취소가 되어야 하는 경우를 들 수 있다. 웹데브 작업 그룹[5]에서는 이러한 웹데브와 관련된 트랜잭션의 처리에 대하여 새로운 메서드를 추가하거나 확장된 버전관리 명세를 통하여 지원하는 방안 등을 논의하고 있다.

(그림 7)은 웹데브 메서드를 수행하는 과정에서 트랜잭션을 처리하는 기본적인 알고리즘을 기술하고 있다. 알고리즘에서는 PostgreSQL 데이터베이스에 대한 질의 수행을 위해 *PQExec()* 함수를 사용하고 있다. 웹데브 메서드의 요청을 받으면 ①에서 트랜잭션을 시작한다. 각 메서드에 따른 자원 속성 정보에 대한 데이터베이스 변경 작업을 수행하는 절차는 먼저 ②에서 로그를 생성하고 ③에서 각 메서드에 따른 DML(Data Manipulation Language) 질의를 수행한 후 ④에서 변경된 값을 로그 정보로 다시 생성한다. 이러한 ②에서 ④까지의 처리가 단일 트랜잭션 구간 내에서 이루어 져야 한다. 이 구간의 작업이 정상적으로 처리되지 못하면 ⑤에서 트랜잭션 복귀 처리를 하고 에러 응답을 돌려준다. 로그 정보가 정상적으로 생성되면 ⑥에서 트랜잭션 승인을 수행하고 자원에 대한 파일 정보를 처리하기 위

한 내부 수행 작업을 진행한다.

속성에 대한 데이터베이스 트랜잭션 처리와 자원에 대한 파일 처리가 분리되어 있기 때문에 성공적으로 처리되지 못하는 비일관성의 문제가 발생할 수 있다. 예를 들어, COPY 메서드에 의해서 자원의 복사를 수행할 때 속성 정보의 복사는 완료되었다 하더라도 자원의 복사가 권한이나 디스크 용량의 문제 등으로 인하여 수행되지 못하는 경우가 발생한다. 물론 이것은 서버 내에서 발생할 수 있는 문제로써 로그 정보나 다른 방법을 통하여 일관성 유지가 가능하다.

이 문제를 해결하기 위하여 본 논문에서는 신뢰성 있는 자원 관리를 위해서 속성에 대한 트랜잭션의 처리와 일관성 관리자를 도입하였다. 일관성 관리자는 트랜잭션 내에서 처리된 작업 수행 전후의 로그 정보를 기반으로 파일이나 디렉터리로 존재하는 자원 정보와 비교하여 비일관성 문제를 해결한다. 만약 동일 자원에 대한 연속적인 변경 작업에 따른 정보의 누락을 방지하기 위해서는 로그 정보에 대한 일관성 관리자의 최종 검토 처리가 끝나지 않은 상태에서는 다른 메서드 작업 처리를 제한할 수도 있다.

```

/* 웹데브 메서드의 요청 처리 */
call dav_get_resource (); /* 자원에 대한 기본 정보를 획득 */
PQExec("BEGIN WORK");
/* 데이터베이스 트랜잭션의 시작 */ -----①
execute log_create(DBConn, Method, Results);
/* 수행 전 로그 생성 */ -----②
switch (Method) { /* 메서드별 데이터베이스 처리 작업 수행 */
-----③
    case PUT : PQExec("INSERT Query");
    case MKCOL : PQExec("INSERT Query");
    case DELETE : PQExec("DELETE Query");
    case MOVE : PQExec("UPDATE Query");
    case COPY : PQExec("INSERT Query");
};
execute log_create (DBConn, Method, Results);
/* 수행 후 로그 생성 */ -----④
if ( error ) {
    PQExec("ROLLBACK WORK");
    /* 데이터베이스 트랜잭션의 롤백 */ -----⑤
    return http_response(error);
} else {
    PQExec("COMMIT WORK");
    /* 데이터베이스 트랜잭션의 승인 */ -----⑥
    call create_temporary_resource(원본 자원 정보);
    /* 임시 자원의 생성 */
};
execute 웹데브 메서드별 내부 수행 코드
if ( error )
    call recovery_temporary_resource(); /* 임시 파일로 복구 */
else
    call delete_temporary_resource(); /* 임시 파일의 삭제 */
return http_response();
    
```

(그림 7) 단일 웹데브 메서드 내의 트랜잭션 처리 알고리즘

4. 시스템 구현

4.1 아파치 모듈의 구현

아파치에서는 웹데브 프로토콜을 지원하고 있는 mod_dav 모듈을 제공한다. mod_dav 모듈은 기본적으로 제공하는 하위 모듈인 mod_dav_fs를 통하여 해당 요청 처리를 수행하도록 되어 있다. 본 논문에서는 파일 시스템 기반의 mod_dav_fs 모듈을 PostgreSQL 데이터베이스 기반의 모듈로 새롭게 구현하여 대치함으로써, 기존의 소스 코드를 변경하지 않고 dav_provider 구조체를 통하여 새로운 인터페이스를 구현할 수 있도록 하였다. 아파치 모듈을 구현하기 위해서는 mod_dav 모듈에서 저장소 인터페이스를 호출하기 위하여 정의한 (그림 8)의 dav_provider 구조체의 후크 함수를 정의하여야 한다. 현재 정의된 구조체는 웹데브의 확장을 고려하여 검색, 버전관리, 바인딩과 같은 명세를 위한 포인터 함수도 함께 정의되어 있다.

```

typedef struct {
    const dav_hooks_repository *repos;
    /* 저장소 인터페이스 후크 함수 */
    const dav_hooks_propdb *propdb;
    /* 속성 메타데이터 처리 후크 함수 */
    const dav_hooks_locks *locks;
    /* 잠금 기능에 대한 후크 함수 */
    const dav_hooks_vsn *vsn;
    /* 버전 관리(DeltaV)에 대한 후크 함수 */
    const dav_hooks_binding *binding;
    /* 자원 바인딩에 대한 후크 함수 */
    const dav_hooks_search *search;
    /* 속성 검색(DASL)에 대한 후크 함수 */
} dav_provider;
    
```

(그림 8) dav_provider 구조체

```

const dav_hooks_repository davinci_hooks_repos = {
    I,
    davinci_get_resource,
    davinci_get_parent_resource,
    davinci_is_same_resource,
    davinci_is_parent_resource,
    ... (생략)
    davinci_create_collection,
    davinci_copy_resource,
    davinci_move_resource,
    davinci_remove_resource,
    davinci_walk,
    davinci_getetag,
};
    
```

(그림 9) dav_hooks_repository 선언 구조체

모듈의 구성은 아파치 모듈 설정을 위한 모듈과 자원의 저장소 관리를 위한 모듈, 웹데브 속성을 관리하기 위한 모듈, 자원의 잠금 정보와 잠금 토큰값을 관리하기 위한 모

들, 그리고 데이터베이스와의 연결과 SQL 처리를 위한 모듈로 구성된다. 저장소 관리 모듈을 자세히 살펴보면, (그림 9)와 같이 *dav_hooks_repository* 선언 구조체에 구현할 함수의 이름을 정의한다. *mod_dav* 모듈에서 자원과 관련된 후크 함수가 호출될 때 (그림 9)에서 정의된 함수가 호출된다.

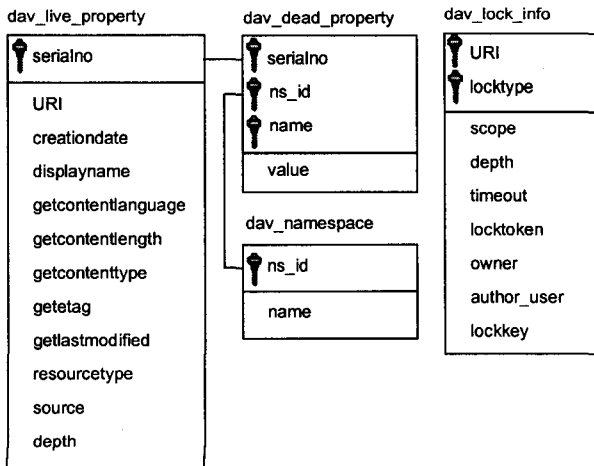
4.2 속성정보 관리를 위한 데이터베이스 스키마

웹데브 속성 정보를 관리하기 위한 데이터베이스 스키마를 (그림 10)과 같이 설계하였다. 테이블의 구성은 기본 속성 정보를 위한 *dav_live_property* 테이블, 사용자가 정의한 속성 정보를 관리하기 위한 *dav_dead_property*와 *dav_namespace* 테이블, 그리고 자원별 잠금 정보를 관리하기 위한 *dav_lock_info* 테이블로 구성된다.

dav_live_property 테이블은 웹데브에서 요구하고 있는 11개의 라이브(live) 속성 정보를 저장한다. 라이브 속성의 값은 서버에 의해서 자동으로 처리된다. 예를 들어, *getcontentlength* 속성은 GET 요청이 있을 때 반환되는 값으로 자원의 소유자가 임의로 설정하는 것이 아니라 웹데브 서버에 의해서 자동으로 계산되어 관리된다[5].

*dav_dead_property*와 *dav_namespace* 테이블은 웹데브의 데드(dead) 속성을 관리한다. 데드 속성은 사용자의 필요에 의해서 자원에 대한 추가적인 속성을 정의한 것으로, 웹데브 서버는 관리만을 담당한다. 속성 정보는 XML 형식의 이름과 값으로 정의되기 때문에 사용자의 속성에 대한 유일성을 보장하기 위한 네임스페이스를 지정할 수 있다. 이러한 사용자의 네임스페이스를 관리하기 위하여 *dav_namespace* 테이블이 필요하다.

dav_lock_info 테이블은 웹데브 잠금 관리를 위한 테이블로서 잠금 상태에 있는 자원의 정보와 잠금 상태, 잠금 토큰 등을 저장한다.

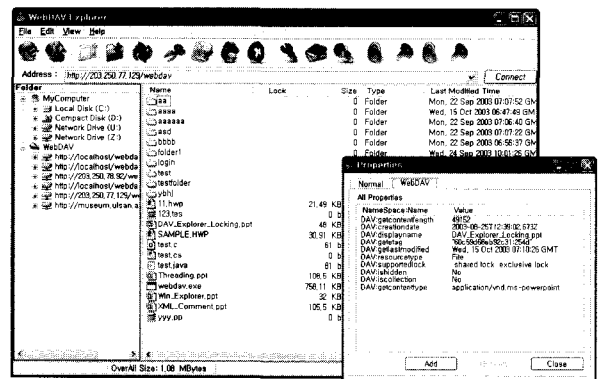


(그림 10) 속성정보를 위한 데이터베이스 스키마

5. 실험 결과

웹데브 서버의 구현 결과로 본 논문에서 설계한 모듈이 *mod_dav* 모듈의 서비스 제공자 모듈에 대한 후크 함수 호출을 받아 웹데브 프로토콜의 명세를 처리하였다. (그림 11)은 웹데브 클라이언트를 통하여 개발된 웹데브 서버에 접속한 화면이다. 웹데브 클라이언트를 통하여 접속할 경우에는 서버의 정보들이 폴더와 파일 형태로 표시되며 데이터베이스를 통하여 속성 정보를 질의하여 추가적으로 보여줄 수 있다. (그림 11)의 웹데브 클라이언트는 본 연구실에서 개발한 닷넷 기반의 웹데브 탐색기로서 마이크로소프트 윈도즈 환경에서 각각의 자원에 적합한 저작 응용 프로그램을 자동으로 실행시켜주고 수정된 정보를 웹데브 서버에 갱신시켜주는 기능을 가지고 있다[13]. 또한 웹 브라우저를 통하여 접속할 경우에는 웹 페이지 형태로 표시가 된다.

본 논문에서 구현한 시스템은 자원의 잠금 기능을 포함한 웹데브 명세의 기본 명세를 지원한다. 주요 베서드 처리 상황을 아파치의 로그 정보를 통하여 정상적으로 수행되는 것을 확인하였으며 다양한 웹데브 클라이언트를 통하여 수행 여부를 확인하였다.



(그림 11) 웹데브 클라이언트를 통한 접속 모습

<표 1>은 아파치 *mod_dav* 모듈의 하위 모듈로 제공되는 서비스 제공자 모듈들 간의 기능을 비교한 표이다. *mod_dav_fs*는 파일 시스템 기반으로 웹데브 기본 명세만을 지원한다. *Catcomb* 모듈은 MySQL 데이터베이스로 자원과 속성을 관리하고 잠금 기능과 *DeltaV*[18], *DASL* 등의 웹데브 주요 명세를 모두 지원하고 있지만, 자원을 데이터 열로써 관리하기 때문에 자원의 관리를 위한 처리 성능이 떨어지고 신뢰성 있는 자원 관리를 위한 트랜잭션 처리를 제공하지 않고 있다. 본 논문에서 구현한 시스템의 주요 장점은 신뢰성 있는 자원 관리를 위한 트랜잭션 처리 기능과 일관성 관리자를 통하여 자원과 속성간의 오류를 제거하는 기능을 제공한다는 것이다.

<표 1> 아파치 mod_dav의 서비스 제공자 모듈간의 기능 비교

웹데브 서버		DAVinci	mod_dav_fs	Catacomb
◦ WebDAV 명세의 지원 (RFC 2518)	◦ 기본 명세 ◦ 잠금 기능	○ ○	○ ○	○ ○
◦ 버전 관리(DeltaV) 지원(RFC 3253)		×	×	△
◦ 속성 검색(DASL) 지원		×	×	○
◦ 자원의 관리 형태		FILE	FILE	DBMS(MySQL)
◦ 속성의 관리 형태		DBMS(PostgreSQL)	FILE(Gnu DBM)	DBMS(MySQL)
◦ 단일 메서드 내의 트랜잭션 지원		○	×	×
◦ Consistency Manager 제공		○	×	×
◦ 운영체제		Linux	Any	Linux

○ : 지원, X : 미지원, △ : 부분지원.

DAVinci 시스템의 정량적 성능 평가를 실험하기 위하여 WebDAV 서버 성능 실험 도구인 Prestan[19]을 이용하였다. Prestan은 웹데브의 주요 항목에 대하여 클라이언트의 입장에서 웹데브 서버로의 요청에 대한 응답 속도를 측정하는 도구이다. 성능 시험을 위한 웹데브 서버의 하드웨어와 소프트웨어의 사양은 <표 2>와 같다.

<표 2> 성능 시험을 위한 웹데브 서버의 사양

CPU	Intel(R) Pentium(R) 4 CPU 2.80GHz
RAM	1024MB
Disk	MAXTOR 6Y080MO, ATA DISK drive 70GB IDE
OS	Linux RedHat 9.0

<표 3> Prestan을 이용한 성능 실험 결과 (단위 : us)

주요 항목	웹데브 서버	DAVinci	Catacomb
ProppatchMult		2466	1965
ProppatchSingle		2466	1967
PropfindDeadMult		2966	2436
PropfindDeadSingle		2964	2464
PropfindLiveMult		4425	4459
PropfindLiveSingle		2970	2786
Put1K		17135	2977
Get1K		2368	2483
Put64K		71025	57796
Get64K		57250	57602
Copy		5916	46607
Move		1666	5440
Delete		1583	3254
MkCol		1750	4032
CopyCol		56151	194497
MoveCol		1755	130291
DeleteCol		1687	88831
Lock		20513	2985
UnLock		15453	2821

<표 3>은 DAVinci 시스템과 Catacomb의 성능을 실험한 결과이다. DAVinci 시스템은 자원과 속성을 파일과 PostgreSQL 데이터베이스로 분리하여 관리하고 있으며, Catacomb는 자원과 속성을 모두 MySQL 데이터베이스로 관리하고 있다. 속성 관리와 관련한 PROPFIND, PROPPATCH 메서드의 처리 성능은 DAVinci가 다소 낮게 나타났다. 이것은 속성을 관리하는 데이터베이스 종류의 차이로 인한 것으로 판단된다. 그러나 자원을 파일 시스템으로 관리하기 때문에 서버의 자원과 컬렉션에 대한 이동, 복사, 삭제의 경우는 DAVinci 시스템이 Catacomb 보다는 매우 뛰어난 성능을 나타내었다.

6. 결 론

DAVinci 시스템은 웹데브의 자원과 속성의 신뢰성 있는 관리를 지원하는 웹데브 서버 시스템이다. DAVinci는 자원을 파일 시스템 형태로 관리하고 자원의 메타데이터에 해당하는 속성을 PostgreSQL 데이터베이스로 관리하도록 개발하였다. 웹데브 자원과 속성의 관리 형태를 파일 시스템과 데이터베이스로 분리함으로써 자원과 속성을 모두 데이터베이스로 관리하는 웹데브 서버 시스템에서 발생할 수 있는 성능 저하 문제를 개선하였다. 또한 데이터베이스를 이용한 속성 관리는 웹데브 서버의 기능 확장 시에도 추가된 명세에 대한 데이터베이스 모델의 추가 설계와 구현을 통한 확장이 용이하다.

DAVinci는 아파치 웹 서버에서 웹데브를 지원하는 mod_dav 모듈의 서비스 제공자 모듈 형태로 구현되었다. DAVinci의 구현 모듈은 mod_dav 모듈에서 정의하고 있는 후크 함수를 구현한 것으로서 모듈의 구성은 아파치 모듈 설정을 위한 모듈과 자원의 저장소 관리를 위한 모듈, 웹데브 속성을 관리하기 위한 모듈, 자원의 잠금 정보를 관리하기 위한 모듈, 그리고 데이터베이스 처리를 위한 모듈로 구성된다.

DAVinci 시스템에서는 웹데브 속성에 대한 데이터베이스

트랜잭션 모델을 지원하고 있다. 그러나 속성에 대한 데이터베이스 트랜잭션 처리와 자원에 대한 파일 처리가 분리되어 있기 때문에 성공적으로 처리되지 못하는 비일관성의 문제가 발생할 수 있다. 이 문제를 해결하기 위하여 본 시스템에서는 신뢰성 있는 자원 관리를 위한 일관성 관리자를 도입하였다. 일관성 관리자는 트랜잭션 내에서 처리된 작업 수행 후의 로그 정보를 기반으로 파일이나 디렉터리로 존재하는 자원 정보와 비교하여 비일관성 문제를 해결한다.

향후 본 시스템의 안정화 작업의 수행과 효과적인 속성 관리를 지원할 수 있도록 서버 시스템을 개선하는 연구를 수행할 계획이다. 또한 확장된 버전관리와 검색 등의 추가 정의된 명세를 지원하는 시스템 및 웹데브 기반의 협업 지원 시스템과 같은 웹데브를 이용한 다양한 정보시스템에 대하여 연구를 수행할 계획이다.

참 고 문 헌

[1] 안건태, 정명희, 이근웅, 문남두, 이명준, "iPlace : EJB 기술을 이용한 웹 기반 협업시스템", 정보처리학회논문지, 제8-D권, pp.735-746, Dec., 2001.

[2] Bentley, R., Horstmann, T., Trevor, J., "The World Wide Web as enabling technology for CSCW : The case of BSCW," Computer Supported Cooperative Work : The Journal of Collaborative Computing, Vol.6, pp.111-134, 1997.

[3] E. James Whitehead, Jr., Meredith Wiggins, "WEBDAV : IETF Standard for Collaborative Authoring on the Web," IEEE Internet Computing, pp.34-40, September/October, 1998.

[4] Jim Whitehead, Yaron Y. Goland, "WebDAV : A network protocol for remote collaborative authoring on the Web," ECSCW'99, 1999.

[5] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, "HTTP Extensions for Distributed Authoring - WEBDAV," RFC 2518, Standards Track, February, 1999.

[6] R. Fielding, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1," RFC 2616, Standards Track, June, 1999.

[7] Microsoft Corporation, "Internet Information Services 5.0 Technical Overview : White Paper," Microsoft Corporation, October, 2001.

[8] Greg Stein, "mod_dav : a DAV module for Apache," http://www.webdav.org/mod_dav/, November, 2001

[9] Software AG, "Tamino WebDAV Server White Paper," Software AG, November, 2001.

[10] Microsoft Corporation, "Microsoft Office Server Extensions : White Paper," Microsoft Corporation, March, 1999.

[11] South River Technologies, WebDrive, [\[drive.com/products/webdrive/\]\(http://drive.com/products/webdrive/\).](http://www.web-</p>
</div>
<div data-bbox=)

[12] Xythos Software Inc., "XYTHOS WFC DATA SHEET," Xythos Software Inc., 2002.

[13] 정혜영, 송동호, 윤병훈, 안건태, 이명준, "닷넷 기반의 Web-DAV Client의 개발", 한국정보과학회, 가을학술발표논문집(III), 제30권 제2호, pp.43-45, 2003.

[14] E-Soft Inc., "Web Server Survey," http://www.security-space.com/s_survey/data/200403/index.html, April, 2004.

[15] Sung Kim, Kai Pan, Elias Sinderson, "mod_dav_dbms: A Database Backed DASL Module for Apache," Department of Computer Science, University of California at Santa Cruz, March, 2002.

[16] Lincoln Stein, Doug MacEachern, "Writing Apache Modules with Perl and C," O'Reilly, 1999.

[17] Karen Schuchardt, James Myers, 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, Vol.5, No.3, pp.287-296, 2002.

[18] C. Kaler, J. Amsden, G. Celmm, B. Cragen, D. Durand, B. Sergeant, E. Whitehead, "Versioning extensions to WebDAV," IETF Internet Draft, January, 1999.

[19] Teng Xu, "Prestan : The Design and Implementation of a WebDAV Server Performance Test Suite," Department of Computer Science, University of California at Santa Cruz, March, 2004.



정 혜 영

e-mail : hyjung@ulsan.ac.kr

1994년 울산대학교 전자계산학과(공학사)

1998년 울산대학교 대학원 전자계학과
(공학석사)

2000년~현재 울산대학교 대학원 컴퓨터
정보통신 공학부 박사과정

1993년~1995년 (주)현대중공업 조선사업부 근무

1997년~2003년 (주)LG CNS 근무(휴직중)

2003년~현재 울산대학교 컴퓨터정보통신공학부 객원교수

관심분야 : WebDAV, 협업지원시스템, 분산 컴퓨팅 등.



안 건 태

e-mail : java2u@ulsan.ac.kr

1999년 울산대학교 전자계산학과(공학사)

2001년 울산대학교 대학원 컴퓨터정보통신
공학부(공학석사)

2001년~현재 울산대학교 대학원 컴퓨터
정보통신공학부 박사과정

관심분야 : 협업지원시스템, 생물정보학, 웹 프로그래밍, 프로그래밍 언어 등



박 양 수

e-mail : yspk56@ulsan.ac.kr

1978년 울산대학교 전자계산학과(학사)
1981년 서울대학교 계산통계학과(석사)
1980년~현재 울산대학교 컴퓨터정보통신
공학부(부교수)
1986년~현재 서울대학교 계산통계학과
박사과정

관심분야 : 분산처리, 컴퓨터알고리즘 등



이 명 준

e-mail : mjlee@ulsan.ac.kr

1980년 서울대학교 수학과(학사)
1982년 한국과학기술원 전산학과(석사)
1991년 한국과학기술원 전산학과(박사)
1993년~1994년 미국 버지니아대학 교환
교수
1982년~현재 울산대학교 컴퓨터정보통신공학부(교수)
관심분야 : 웹기반 정보시스템, 프로그래밍언어, 분산 프로그래
밍, 생물정보학