

다중 응용을 위한 멀티 디렉토리 모델링 및 구현 방안

正會員 金榮俊*, 林宰弘**

Modelling and Implementation Scheme of a Multi Directory for Multiple Applications

Young-Jun Kim*, Jae-Hong Yim** *Regular Members*

요 약

본 논문에서는 분산 환경에서 다중 응용의 위치 정보 및 부가 정보를 제공하는 멀티 디렉토리 시스템의 모델링 및 구현 방안을 제시한다.

멀티 디렉토리를 위한 확장된 디렉토리 스키마(schema), 확장된 디렉토리 트리 구조(DIT: Directory Information Tree) 모델을 제안하고, 구현 방안을 제시하기 위하여 문서 화일링 및 검색(DFR: Document Filing and Retrieval) 응용에 적용한 구현 예를 보인다. 특히 디렉토리의 확장과 더불어 DFR 응용을 사용할 때 자동으로 멀티 디렉토리를 이용할 수 있도록 하기 위하여 DFR과 멀티 디렉토리 상호연동을 위한 구성 모델 및 두 응용 동작들 사이의 연계 메카니즘 등도 기술한다.

분산 환경하에서 DFR과 ISODE(ISO Development Environment) 디렉토리 시스템인 QUIPU 8.0을 이용하여 구현한 멀티 디렉토리 사이의 연동 시험을 하고, 본 논문의 결과를 기반으로 다양한 응용 시스템의 객체 탐색을 위한 확장된 디렉토리 시스템 모델을 제시한다.

ABSTRACT

This paper presents a modelling and implementation scheme of a multi directory which provides location and supplementary information of multiple applications in distributed environments.

An extended directory schema and extended DIT(Directory Information Tree) model is proposed for the multi directory. For implementation scheme, an implementation example adapted for the DFR(Document Filing and Retrieval) is shown. In addition, a configuration model for interworking between the DFR and the multi directory, an association mechanisms between two application's operations are also described.

*대전전문대 비서행정과

**한국해양대학교 전자통신공학과

論文番號:96230-0805

接受日字:1996年 8月 5日

In a distributed environment, the interworking between the DFR and a multi directory implemented by using QUIPU 8.0 of ISODE(ISO Development Environment) directory system is tested. Based on the result of this paper, an extended model of the directory system is proposed for providing search operation of various applications' objects.

I. 서 론

분산 사무 환경은 PC, 워크스테이션 및 다른 여러 정보기기를 통신망으로 연결하여 분산된 자원의 효율적 공유를 가능케 한다. 따라서 분산 사무 환경에서는 PC와 워크스테이션들의 통신망 구축에 의한 정보의 공유 및 시스템의 효율적인 사무 처리기능이 제공되어야 하는데, 이러한 분산 사무 환경에서 사용자가 사무 정보를 효율적이고 편리하게 처리하기 위해서는 데이터 전송 기능, 개인 상호간의 메시지 교환 기능, 디렉토리 서비스 기능, 문서의 화일링과 검색 기능 등의 지원을 필요로 한다.

이러한 요구에 따라 국제 표준화 기구(ISO: International Standards Organization)에서는 분산 사무 처리에 필요한 클라이언트-서버 모델의 여러 응용 시스템들을 통합 처리할 수 있는 분산 사무응용 모델(DOAM: Distributed Office Application Model)^(1, 2, 8)을 제안하고 이에 대한 표준화 작업을 진행하고 있으며, 응용 시스템으로써 메시지 처리 시스템(MHS: Message Handling System)⁽³⁾, 디렉토리 시스템(directory system)⁽⁴⁾, 문서 화일링 및 검색(DFR: Document Filing and Retrieval) 시스템^(5, 6, 9), 문서 인쇄 응용(DPA: Document Printing Application)⁽⁷⁾ 등을 규정하고 있다.

디렉토리 시스템은 네트워크 사용자들의 이름, 네트워크 주소, 소속기관 등 통신 서비스에 필요한 여러 정보들을 체계적이고 효율적으로 저장하여 검색 및 관리 서비스를 제공하는 시스템이며 메시지 처리 시스템에서 사용자의 네트워크 주소를 검색하기 위한 도구로써 주로 사용되고 있다. 그러나 여러 응용 시스템들이 혼재되어 있는 분산 환경에서는 사용자의 다양한 요구사항을 충족시킬 수 있는 확장된 디렉토리 서비스가 제공되어야 한다.

본 논문에서는 분산 환경에서 다중 응용의 위치 정보 및 부가 정보를 제공하는 멀티 디렉토리 시스템의 모델링 및 구현 방안을 제시한다. 멀티 디렉토리를 위한 확장된 디렉토리 스키마(schema), 확장된 디렉토

리 트리 구조(DIT: Directory Information Tree) 모델을 제안하고, 구현 방안을 제시하기 위하여 문서 화일링 및 검색(DFR: Document Filing and Retrieval) 응용에 적용한 구현 예를 보인다. 특히 디렉토리의 확장과 더불어 DFR 응용을 사용할 때 자동으로 멀티 디렉토리를 이용할 수 있도록 하기 위하여 DFR과 멀티 디렉토리 상호연동을 위한 구성 모델 및 두 응용 동작들 사이의 연계 메카니즘 등도 기술한다.

분산 환경하에서 DFR과 ISODE(ISO Development Environment) 디렉토리 시스템인 QUIPU 8.0을 이용하여 구현한 멀티 디렉토리 사이의 연동 시험을 하고, 본 논문의 결과를 기반으로 다양한 응용 시스템의 객체 탐색을 위한 확장된 디렉토리 시스템 모델을 제시한다.

본 논문의 II장에서는 멀티 디렉토리 모델에 대하여, III장에서는 멀티 디렉토리 구현 방안에 대하여 기술한다. 마지막으로 IV장에서는 결론을 서술한다.

II. 멀티 디렉토리 모델

1. 기능 모델

디렉토리 시스템⁽⁴⁾은 네트워크에 연결되어 있는 정보기거나 사용자들의 이름, 네트워크 주소, 소속기관 등 통신 서비스에 필요한 여러 정보들을 체계적이고 효율적으로 저장하여 검색 및 관리 서비스를 제공하는 시스템이며, ITU-T(International Telecommunication Union)에서는 디렉토리 시스템에 대한 국제 표준으로써 X.500 권고안을 발표하였다. 디렉토리 시스템이 제공하는 기본적인 서비스로는 송·수신자명 존재 검증, 송·수신자에 대한 주소 찾기, 이름 일부로 송·수신자명 찾기, 사용자에게 디렉토리 전체 열람 혹은 선택 열람 권한 허용, 디렉토리를 최신 상태로의 유지 등이 있다.

멀티 디렉토리 기능 모델은 그림 I과 같이 멀티 디렉토리 사용자 처리기(MDUA: Multi Directory User Agent)와 멀티 디렉토리 시스템 처리기(MDSA: Multi

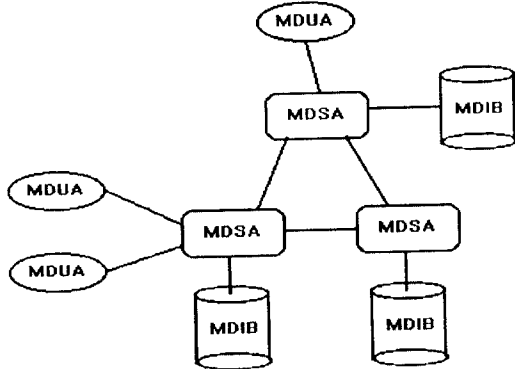


그림 1. 멀티 디렉토리 기능 모델
Fig. 1 Multi directory functional model

Directory System Agent)로 구성된다.

그림 1에서 MDUA는 사용자의 요청을 MDSA에 전달해 주고, MDSA로부터 얻은 정보를 사용자에게 알려주는 응용 프로세스이며, MDSA는 분산된 디렉토리 정보를 저장, 관리해 준다. 각각의 MDSA는 디렉토리에 등록된 로컬(local) 정보를 관리하고, 멀티 디렉토리 정보 베이스(MDIB: Multi Directory Information Base)를 형성한다. MDSA는 MDUA로부터 요청받은 서비스에 응답하기 위하여 우선적으로 로컬 데이터베이스에 액세스하며, 결과를 얻지 못한 경우에는 다른 MDSA에 액세스하게 된다.

2. 확장된 디렉토리 스키마

디렉토리 스키마는 디렉토리 정보 트리(DIT: Directory Information Tree) 구조, 객체 클래스(object class), 속성 타입 및 구문(syntax)에 대한 정의 및 규칙을 표기하는 집합이며, 이는 부적합한 객체들의 생성, 객체에 해당되지 않는 속성 집합의 추가 및 갱신, 속성 타입에 어긋난 속성값들의 할당 등을 제어, 관리함으로써 효율적인 디렉토리를 유지하는데 중요한 기능을 담당한다.

현재 디렉토리 시스템의 표준안은 사용자 정보를 관리하기 위한 스키마에 대해서만 규정되어 있으므로, 다중 응용들의 속성을 관리하기 위해서는 디렉토리 스키마를 확장하여야 한다. 그러나 디렉토리 스키마의 확장은 표준안을 벗어나지 않는 범위에서 이루어져야 하며, 본 논문에서의 확장된 디렉토리 스키마도

표준안에 기반을 두고 설계하도록 한다.

2.1 확장된 DIT 구조

기존 디렉토리 시스템의 DIT를 확장하여 다중 응용들의 속성을 관리하게 할 경우 확장된 DIT 구조는 그림 2에 보인 바와 같다.

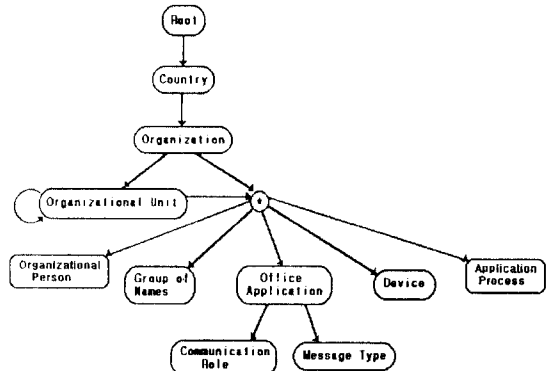


그림 2. 확장된 DIT 구조
Fig. 2 Extended DIT configuration

그림 2에서 root→country→organization→organizational unit→organizational person 트리 구조는 기존 디렉토리의 DIT 구조이며, 여기에 GroupOfNames, OfficeApplication, Application Process, Device 등의 객체 클래스를 정의하여 기존 DIT 구조를 확장하였다. 그림에서와 같이 추가되는 객체 클래스는 표준안의 범위를 벗어나지 않도록 기존 DIT 구조의 organization 혹은 organizational unit 클래스의 하부에 정의하도록 한다.

2.2 객체 클래스

확장된 디렉토리 스키마에서는 그림 2에서와 같이 X.521에 정의된 표준 객체 클래스인 country, organization, organizationalUnit 클래스를 그대로 이용하고, organization 클래스 아래 그룹통신에 사용할 수 있는 GroupOfNames와 분산 환경에서 여러가지 공유장치 지정할 수 있는 Device 객체 클래스를 정의함으로써 네트워크 사용자에 대한 정보외에 다양한 객체 관리에 디렉토리를 이용할 수 있도록 하고, 사무응용 시스템을 위한 OfficeApplication 클래스와 OA 클래스

스내에 문서나 메시지 타입, 기능 역할을 나타내는 클래스를 정의함으로써 기존 디렉토리를 확장 사용할 수 있도록 하였다. 새로이 정의된 객체 클래스는 다음과 같다.

```
groupOfNames OBJECT-CLASS
  SUBCLASS OF Top
  MUST CONTAIN {
    commonName,
    member }
  MAY CONTAIN {
    description,
    organizationName,
    owner,
    seeAlso,
    businessCategory }
```

```
officeApplication OBJECT-CLASS
  SUBCLASS OF Top
  MUST CONTAIN {
    commonName }
  MAY CONTAIN {
    description,
    localityName,
    organizationalUnitName,
    seeAlso,
    processFile,
    procedureFile,
    procedureSystem }
```

```
applicationProcess OBJECT-CLASS
  SUBCLASS OF Top
  MUST CONTAIN {
    commonName }
  MAY CONTAIN {
    description,
    localityName,
    organizationalUnitName,
    processFile,
    seeAlso }
```

위에서 정의한 객체 클래스에 대한 속성 값의 예를 들면 다음과 같다.

```
- officeApplication
commonName: editor
description: UNIX text editor
localityName: Server A
processFile: /usr/bin/vi
```

```
- applicationProcess
commonName: Print service
description: Print service for high quality
localityName: Server B
processFile: /usr/bin/lpd
seeAlso: Device: printer-07
```

2.3 속 성

새로이 정의된 객체 클래스를 구성하는 속성 중 주요 속성에 대한 내용과 구문은 다음과 같다.

```
- member: 그룹 멤버들의 이름을 지정
  member ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
    distinguishedNameSyntax }

- localityName: 서비스 혹은 프로세스를 포함하고
  있는 시스템 지정
  localityName ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
    caseIgnoreStringSyntax }

- processFile: 서비스를 제공하는 프로세스 파일의
  이름을 지정
  processFile ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
    caseIgnoreStringSyntax }
```

Ⅲ. 멀티 디렉토리 구현 방안

본 장에서는 기존의 디렉토리를 확장한 멀티 디렉토리의 구현 방안을 제시하기 위하여 DFR 응용에 적

용한 구현 예를 보인다. 특히 디렉토리의 확장과 더불어 DFR 응용을 사용할 때 자동으로 멀티 디렉토리를 이용할 수 있도록 하기 위하여 DFR 시스템과 멀티 디렉토리 시스템 상호연동을 위한 구성 모델 및 두 응용 동작들 사이의 연계 메카니즘 등도 기술한다.

DFR 시스템^{5,6)}은 디스크 용량이 제한된 PC나 워크스테이션에서 원격 동작(remote operation)으로 대용량의 서버에 화일링되어 있는 문서를 관리하는 응용 시스템으로써, 문서 내용에 무관한 통신 프로토콜을 규정하여 일반 텍스트 문서나 멀티미디어 문서 또는 개방 문서 구조(ODA: Open Document Architecture) 표준안에 따른 문서를 속성을 이용하여 화일링하거나 검색, 수정, 삭제 등의 동작을 수행한다. 즉 MHS가 OSI(Open Systems Interconnection) 계층 구조를 따르는 전자 우편 시스템의 국제 표준안으로 제안되었듯이 DFR 시스템은 일종의 분산 화일 시스템에 대한 국제 표준이라 할 수 있으나, 화일 저장소(file store)의 구조에 대한 표준화보다는 원격 동작 서비스 요소의 표준화에 주안점을 두고 있다. DFR 시스템은 사용자에게 서비스를 제공하기 위하여 하나의 서버 안에 하나의 문서 저장기(DS: Document Store)를 두고 있으며, 일반 화일 시스템 구조와 같이 트리 구조로 구성된다.

1. 기본 개념

DFR 시스템의 문서 저장기에 화일링되는 DFR 객체는 속성부와 내용부로 구성되는데, 이를 화일 시스템에 화일링하는 경우 i) 속성부와 내용부를 하나의 화일에 저장하는 방법, ii) 내용부와 속성부를 별개의 화일에 저장하는 방법, iii) 화일 시스템을 확장하여 화일 시스템의 화일 속성을 수정하는 방법 세 가지가 가능하나, DFR 시스템의 이식성을 높게 하기 위해서는 i), ii)의 방법이 바람직하다. 두 가지 방법 중, DFR 시스템과 디렉토리 시스템과의 연동을 위하여 i)의 방법보다는 내용부와 속성부를 별개의 화일에 저장하는 ii)의 방법이 적절하며, DFR 시스템의 속성부를 멀티 디렉토리의 확장된 DIT에서 관리하도록 한다. DFR 시스템의 내용부와 속성부는 고유식별자(UPI: Unique Permanent Identifier)와 같은 고유속성을 이용하여 매핑되도록 하며, DFR 속성부는 문서 저장기 구조와 논리적으로 동일한 트리 구조를 이루도록 구

성한다.

2. 시스템 구성 모델

멀티 디렉토리 시스템과 DFR 시스템과의 연동을 위한 시스템 구성 모델은 그림 3과 같다.

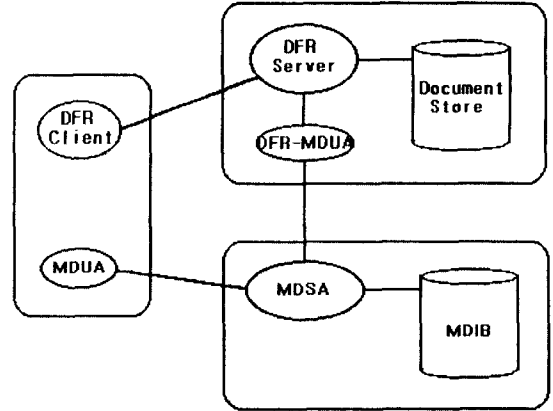


그림 3. 시스템 구성 모델
Fig. 3 System configuration model

그림 3과 같이 멀티 디렉토리 시스템과 DFR 시스템을 기존 사용 방식대로 각각의 클라이언트를 이용하여 개별적으로 사용할 수도 있으며, DFR 시스템의 탐색 동작 사용시 DFR 서버와 MDSA간 인터페이스를 구성하는 DFR-MDUA에 의하여 자동적으로 멀

표 1. DFR, 디렉토리 동작 사이의 매핑관계
Table 1. Mapping relationship between DFR and directory operations

| 기 능 | DFR 동작 | 디렉토리 동작 | 기 능 |
|------------|--------|-------------|--------|
| 연계 설정 | bind | bind | 연계 설정 |
| 연계 해제 | unbind | unbind | 연계 해제 |
| DFR-객체의 생성 | create | AddEntry | 속성의 추가 |
| DFR-객체의 복사 | copy | AddEntry | 속성의 추가 |
| DFR-객체의 삭제 | delete | RemoveEntry | 속성의 제거 |
| DFR-객체의 수정 | modify | ModifyEntry | 속성의 수정 |
| DFR-객체의 탐색 | search | search | 속성의 탐색 |
| DFR-객체의 읽기 | read | read | 속성값 읽기 |
| DFR-객체의 열거 | list | list | 속성의 열거 |

티 디렉토리를 이용할 수 있게 함으로써, DFR 사용 시에는 사용자에게 투명한(tranparent) 사용 환경을 제공하도록 한다.

기본적으로 DFR 탐색 동작 사용시 DFR 서버는 멀티 디렉토리를 이용하여 사용자가 원하는 DFR 객체에 대한 속성을 알려주지만, 이를 위해서는 생성(create), 복사(copy), 수정(modify) 등과 같이 DFR 객체의 속성을 변화시키는 DFR 동작들이 디렉토리 동작과 연계가 이루어져 디렉토리내의 DFR 속성부와 DFR 문서 저장기가 동일한 트리 구조를 유지하도록 하여야 한다. 이를 위한 DFR 동작과 디렉토리 동작 사이의 매핑관계를 표 1에 나타내었다.

3. 확장된 디렉토리 스키마(schema)

3.1 확장된 DIT 구조

기존 디렉토리 시스템의 DIT를 확장하여 DFR 속

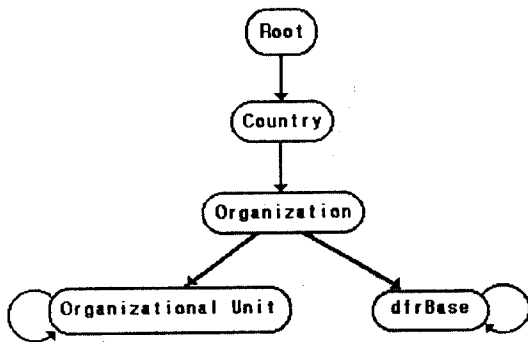


그림 4. 확장된 DIT 구조
Fig. 4 Extended DIT configuration

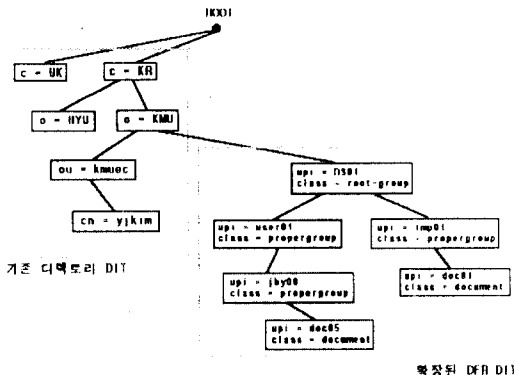


그림 5. DIT 구성 모델
Fig. 5 DIT configuration model

성을 관리하게 할 경우 확장된 DIT 구조와 구성 예는 각각 그림 4와 그림 5에 보인 바와 같다. II장에서 기술하였듯이 표준안의 범위를 벗어나지 않도록 하기 위하여 dfrBase 클래스를 organization 클래스 하부에 정의하였다.

3.2 객체 클래스

dfrBase 객체 클래스는 다음과 같이 정의된다.

dfrBase OBJECT-CLASS

SUBCLASS OF Top

MUST CONTAIN {

- dfr-upi,
- dfr-object-class,
- dfr-title,
- dfr-pathname,
- version-name,
- create-date-and-time,
- created-by,
- dfr-access-list }

MAY CONTAIN {

- owner,
- document-type,
- subject }

3.3 속 성

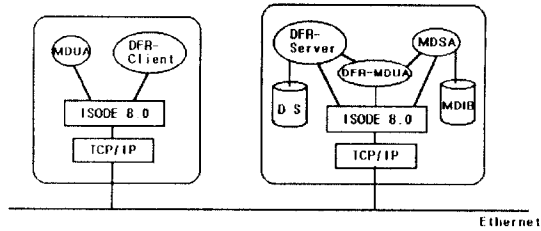
dfrBase 객체 클래스를 구성하는 속성 중 주요 속성에 대한 내용은 다음과 같다.

- dfr-upi: 고유식별자로써 DFR 서버에 의해 할당되며 DFR 객체를 고유하게 지정하는데 사용된다.
- dfr-object-class: DFR 객체의 종류를 지정하는데 사용된다.
- dfr-title: DFR 객체의 이름을 나타내며, 사용자에 의해 명시된다.
- dfr-pathname: 문서 저장기내에서의 DFR 객체의 위치를 나타내며, 루트 그룹으로부터 객체까지의 경로명이 지정된다.
- owner: DFR 객체의 소유자를 나타낸다.

4. 구현 및 시험

4.1 구현 환경

멀티 디렉토리의 구현 환경은 그림 6과 같이 Ethernet LAN으로 연결되어 있는 두 대의 SUN 워크스테이션을 각각 클라이언트와 서버로서 사용하며, ISODE 개발툴⁽¹⁾을 사용하도록 한다. DFR 시스템으로는 ISODE를 이용하여 개발된 HY-DFR 시스템⁽²⁾을 사용하고, ISODE 디렉토리 시스템인 QUIPU 8.0 소스 분석을 통해 확장 구현하여 HY-DFR 시스템과 연동시킨다. 그림에서는 DFR 서버와 MDSA가 한 시스템에 존재하지만, DFR-MDUA는 호스트 이름을 가지고 MDSA에 접속을 하므로 MDSA가 다른 시스템에 존재하여도 무방하다.



DS: Document Store

그림 6. 구현 환경
Fig. 6 Implementation environment

4.2 멀티 디렉토리와 DFR 동작 사이의 연계 메카니즘

본 논문에서의 멀티 디렉토리는 기존 사용 방식대로 디렉토리 클라이언트인 MDUA를 통하여, 기존의 사용자에 대한 속성뿐 아니라 새로이 추가된 DFR 객체에 대한 속성도 검색할 수 있다. 또한 DFR 시스템 사용시에는 DFR 서버와 디렉토리 서버인 MDSA간 인터페이스 역할을 하는 DFR-MDUA에 의하여 자동적으로 멀티 디렉토리를 이용할 수 있게 함으로써, 사용자에게 투명한 사용 환경을 제공하게 된다. 따라서 DFR 사용시, DFR 클라이언트와 서버간 bind 동작이 수행되면 그와 동시에 DFR 서버는 DFR-MDUA를 통하여 MDSA와 bind 동작을 자동적으로 수행하게 된다. 그러므로 DFR 객체의 속성을 변화시키는 DFR 동작이 요청되면 DFR-MDUA를 통한 디렉토리 동작과 연계가 이루어짐으로써 DFR 문서 저장기와 디렉토리내의 DFR 속성간 일관성이 유지될 수 있다.

멀티 디렉토리에 DFR 속성들은 디렉토리의 EDB (Entry Data Block) 구조로 저장, 관리되는데, 이를 DFR-EDB라 명명하였으며, DFR 객체의 속성을 변화시키는 생성, 삭제, 복사 등과 같은 동작이 수행되면, DFR 속성은 엔트리 단위로 DFR-EDB가 수정된다. 본 절에서는 DFR 동작과 디렉토리 동작 사이의 주요 연계 메카니즘 중에서 생성 동작에 대하여만 기술하도록 한다.

DFR 동작 중 생성 동작은 DFR 객체를 새로 생성하는 동작이므로, 디렉토리에 속성을 추가하는 Add-Entry 동작과 연계가 이루어져야 하며, 연계 메카니즘은 그림 7과 같다. 그림에서와 같이 DFR 사용자가 사용자 정의 속성과 내용을 작성하여 객체 생성을 요청하면, DFR 서버는 사용자 이름과 접근권한을 검사한 후 속성과 내용을 분리하고 서버 정의 속성을 생성하게 된다. DFR 서버는 객체의 속성집합을 DFR-MDUA를 통하여 확장 디렉토리에 보내어 DFR-EDB에 엔트리를 추가하고 결과를 받으면 문서 저장기에 객체를 생성한 후 사용자에게 결과를 알려주게 된다.

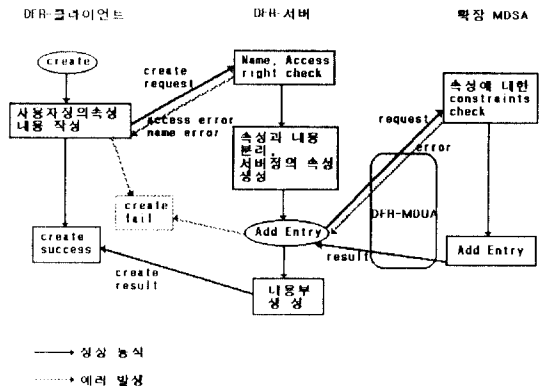


그림 7. 생성 동작의 연계 메카니즘
Fig. 7 Association mechanism in create operation

4.3 시험 및 고찰

본 논문에서 구현한 멀티 디렉토리의 검증은 위하여, 디렉토리 클라이언트인 MDUA를 이용하여 DFR 객체를 탐색하는 방법과 DFR 시스템 사용시 탐색 동작을 이용하여 탐색하는 방법 두 가지 동작 시험을

하였으며, 동일한 결과를 얻음으로써 디렉토리내의 DFR 속성에 대한 일관성과 사용방법에 대한 유연성을 확인하였다. 먼저 QUIPU의 DUA인 'dish'를 이용하여 기존에 디렉토리 시스템을 사용하는 방법 그대로 DFR 객체를 탐색하였으며, DFR 시스템 사용시에는 탐색 동작시 DFR-MDUA에 의하여 자동적으로 확장 디렉토리를 이용하게 함으로써 투명성을 제공하였다.

DFR 시스템 표준안에는 DFR 객체의 탐색 동작시 디렉토리 시스템을 이용하도록 언급은 되어 있으나 구체적인 방법 및 해결책은 제시되지 않고 있다. 따라서 본 논문에서는 DFR 시스템 탐색 동작에 디렉토리 시스템을 이용할 수 있는 하나의 방안을 제시하였으며, 본 논문의 결과를 이용하여 다양한 사무응용 시스템의 객체 탐색에 디렉토리 시스템을 이용할 수 있는 확장 모델을 제안한다.

본 논문에서 구현한 멀티 디렉토리 시스템은 기존 DUA 사용방법 그대로 사용할 수도 있으며, DFR 사용중에는 탐색 동작 사용만으로 디렉토리를 이용할 수 있다는 유연성이 제공된다.

IV. 결 론

본 논문에서는 분산 환경에서 사용자의 다양한 요구사항을 만족시킬 수 있도록 다중 응용의 위치 정보 및 부가 정보를 제공하는 멀티 디렉토리 시스템의 모델링 및 구현 방안을 제시하였다. 멀티 디렉토리를 위한 확장된 디렉토리 스키마(schema), 확장된 디렉토리 트리 구조(DIT: Directory Information Tree) 모델을 제안하고, 구현 방안을 제시하기 위하여 문서 화일링 및 검색(DFR: Document Filing and Retrieval) 응용에 적용한 구현 예를 보였으며, 특히 디렉토리의 확장과 더불어 DFR 응용을 사용할 때 자동으로 멀티 디렉토리를 이용할 수 있도록 하기 위하여 DFR과 멀티 디렉토리 상호연동을 위한 구성 모델 및 두 응용 동작들 사이의 연계 메카니즘 등도 기술하였다.

분산 환경하에서 DFR과 ISODE(ISO Development Environment) 디렉토리 시스템인 QUIPU 8.0을 이용하여 구현한 멀티 디렉토리 사이의 연동 시험을 하고, 본 논문의 결과를 기반으로 다양한 응용 시스템의 객체 탐색을 위한 확장된 디렉토리 시스템 모델을

제시하였다.

향후 연구로는 다중 응용들이 혼재함으로써 일어나는 스키마 충돌, 다중 응용 속성의 효율적인 운용 모델 제시 등이 필요하며, LDAP(Lightweight Directory Access Protocol) 개념을 도입하여 인터넷상에서도 활용할 수 있는 방안이 추진되어야 할 것이다.

참 고 문 헌

1. ISO/IEC 10031-1, Distributed Office Application Model(Part 1:General Model), 1991.
2. ISO/IEC 10031-2, Distirbuted Office Application Model(Part 2:Distinguished Object Reference and Associated Procedures), 1991.
3. CCITT Recommendation X.400, Message Handling System, 1992.
4. ISO 9594/CCITT Recommendation X.500, The Directory, 1992.
5. ISO/IEC 10166-1, Document Filing and Retrieval (Part 1:Abstract Service Definition and Procedure), 1991.
6. ISO/IEC 10166-2, Document Filing and Retrieval (Part 2:Protocol Specification), 1991.
7. ISO/IEC 10175-1, Document Printing Application (Part 1:Abstract Service Definition and Procedure), 1991.
8. P. Chilton, "Current status of DOAM," Intl. Open Systems, Blenheim Online Publications, pp. 207-218, 1989.
9. Yuji Miyawaki, "A Feasibility Study on DFR," 일본정보처리학회 멀티미디어 통신과 분산 처리 연구회지, Vol. 47, No. 7, pp. 1-8, 1990.
10. CCITT Recommendation X.208, Specification of ASN.1, 1988.
11. Marshall T. Rose, The ISODE:User's Manual (Version 8.0), Performance Systems International Inc., 1992.
12. 임재홍, 박용진, "OSI 환경에서 문서 화일링 및 검색 시스템의 설계 및 구현," 대한전자공학회 논문집, 제 31 권 B 편, 제 2 호, pp. 10-20, 1994.
13. K. Obraczka, "Internet Directory Service," IEEE

Computer, pp. 8-22, Sep. 1993.

14. T. Johannsen, G. Mansfield, "Optimizing Document Retrieval by Using the X.500 Directory:Soft Pages project," Draft Document OSI-DS 39, 1992.



김 영 준(Young-Jun Kim) 정회원

1986년 2월:한양대학교 전자공학과 졸업(공학사)

1990년 8월:한양대학교 대학원 전자공학과 졸업(공학석사)

1994년 2월:한양대학교 대학원 전자공학과 박사과정

수료

1996년 3월~현재:대전전문대학 비서행정과 전임강사

※주관심분야:분산처리, 이동통신



임 재 홍(Jac-Hong Yim) 정회원

1986년 2월:서강대학교 전자공학과 졸업(공학사)

1988년 8월:한양대학교 대학원 전자공학과 졸업(공학석사)

1995년 2월:한양대학교 대학원 전자공학과 졸업(공학

박사)

1995년 3월~현재:한국해양대학교 전자통신공학과 조교수

※주관심분야:분산처리, 그룹웨어, 통신망관리