

DFR 속성 관리를 위한 확장된 디렉토리 시스템의 설계 및 구현

임재홍[†] · 김영준^{††}

요약

본 논문에서는 문서 파일링 및 검색(DFR: Document Filing and Retrieval) 시스템에서 DFR 객체의 속성(attribute)을 체계적으로 관리하여 DFR 객체의 효율적인 탐색(search) 기능을 제공하기 위한 확장된 디렉토리 시스템의 설계 및 구현에 관하여 논한다. 이를 위하여 DFR 시스템과 디렉토리 시스템 상호연동을 위한 구성 모델 및 두 응용 시스템 동작들 사이의 연계 메카니즘, 확장된 디렉토리 스키마(schema), 확장된 디렉토리 트리 구조(DIT: Directory Information Tree)를 설계, 구현한다.

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

Design and Implementation of an Extended Directory System for Management of the DFR Attributes

Jae-Hong Yim[†] · Young-Jun Kim^{††}

ABSTRACT

This paper presents a design and implementation of an extended directory system in order to manage and provide an effective search operation of the DFR(Document Filing and Retrieval) Object's attributes. For this, the design and implementation of a configuration model for interworking between the DFR and directory systems, an association mechanisms between two application's operations, extended directory schema and extended DIT (Directory Information Tree) are described.

In a distributed environment, the interworking between the DFR and directory system 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.

1. 서론

분산 사무 환경은 PC, 워크스테이션 및 다른 여러

정보기기들을 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 위

[†] 정 회 원: 한국해양대학교 전자통신공학과
^{††} 정 회 원: 대전전문대 비서행정과
논문접수: 1996년 8월 9일, 심사완료: 1996년 9월 17일

해서는 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 도구로서 주로 사용되고 있다.

본 논문에서는 문서 화일링 및 검색 기능을 제공하는 DOAM 응용 시스템의 하나인 DFR 시스템에서 DFR 객체의 속성(attribute)을 체계적으로 관리하여 DFR 객체의 효율적인 탐색(search)을 위한 확장된 디렉토리 시스템의 설계 및 구현에 관하여 논한다. DFR 시스템 표준안에는 DFR 객체의 탐색 동작시 디렉토리 시스템을 이용하도록 언급은 되어 있으나 구체적인 방법 및 해결책은 제시되지 않고 있다. 따라서 본 논문에서는 DFR 속성 관리를 위한 확장된 디렉토리 시스템 모델 및 DFR 시스템과의 연동을 위한 각 동작들 사이의 연계 메카니즘을 설계하고, 이를 ISODE(ISO Development Environment) 디렉토리 시스템인 QUIPU 8.0을 이용하여 구현, 시험한다.

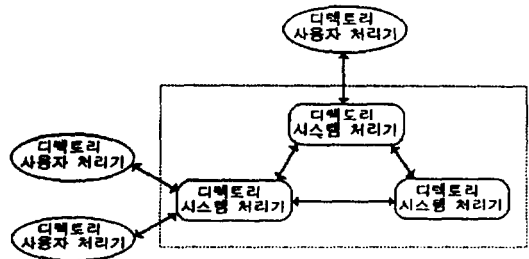
본 논문의 2장에서는 디렉토리 시스템과 DFR 시스템의 개요에 대하여 간략히 기술하고, 3장에서는 확장 디렉토리 시스템의 설계 방법에 대하여 기술한다. 4장에서는 구현 및 동작시험에 대하여, 마지막으로 5장에서는 결론을 서술한다.

2. 디렉토리 및 DFR 시스템의 개요

2.1 디렉토리 시스템

디렉토리 시스템[4]은 네트워크에 연결되어 있는 정보기구나 사용자들의 이름, 네트워크 주소, 소속기관 등 통신 서비스에 필요한 여러 정보들을 체계적이고 효율적으로 저장하여 검색 및 관리 서비스를 제공하는 시스템이며, ITU-T에서는 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.

디렉토리 시스템은 OSI 참조 모델의 응용 계층에 해당하며, 그 기능 모델은 (그림 1)과 같이 디렉토리 사용자 처리기(DUA: Directory User Agent)와 디렉토리 시스템 처리기(DSA: Directory System Agent)로 구성된다.



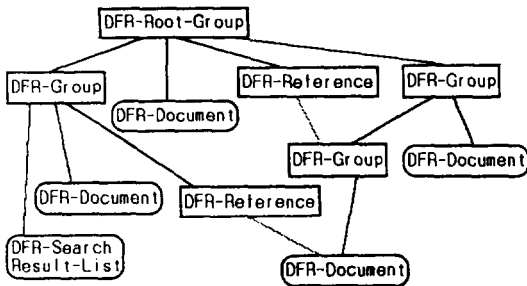
(그림 1) 디렉토리 기능 모델
(Fig. 1) Directory functional model

(그림 1)에서 DUA는 사용자의 요청을 DSA에 전달해 주고, DSA로부터 얻은 정보를 사용자에게 알려 주는 응용 프로세스이며, DSA는 분산된 디렉토리 정보를 저장, 관리해 준다. 각각의 DSA는 디렉토리에 등록된 로컬(local) 사용자들에 대한 정보를 관리하고, 디렉토리 정보 베이스(DIB: Directory Information Base)를 형성한다. DSA는 DUA로부터 요청받은 서비스에 응답하기 위하여 우선적으로 로컬 데이터베이스에 액세스하며, 결과를 얻지 못한 경우에는 다른 DSA에 액세스하게 된다.

2.2 DFR 시스템

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 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 서버안에 하나의 문서 저장기(DS: Document Store)를 두고 있는데, 그 구조는 (그림 2)와 같다.



(그림 2) DFR 문서 저장기의 구조
(Fig. 2) DFR document store structure

(그림 2)에서 보는 바와 같이 문서 저장기를 이루는 DFR 객체(DFR Object)는 트리(tree) 구조로 이루어져 있다. 각 DFR 객체는 속성부(attribute)와 내용부(content)로 구성되고, DFR 객체를 고유하게 지정하는데 사용되는 고유식별자(UPI: Unique Permanent Identifier)를 가지고 있다. DFR 객체의 종류는 다음과 같다.

- 루트 그룹(root group): 문서 저장기의 최상위에 위치하며 모든 DFR 객체를 가리키고 있는 포인터의 집합으로 이루어져 있다.

- 일정 그룹(proper group): 루트 그룹내에서 여러 DFR 객체를 포함하는 단위로서, 일정 그룹 자체의 정보를 나타내는 속성부와 그룹내에 포함되어 있는 DFR 객체들의 리스트를 저장하는 내용부로 이루어진다. 일정 그룹에 접근할 수 있는 사용자들을 제한함으로써 그룹내의 DFR 객체는 사용자가 받지 못한 사용자로부터 보호받을 수 있다.

- 참조(reference): 참조는 문서 저장기내의 루트 그룹과 또 다른 참조를 제외한 DFR 객체를 가리키는 포인터로서, DFR 객체를 링크시켜서 그에 대한 정보를 저장하고 있다.

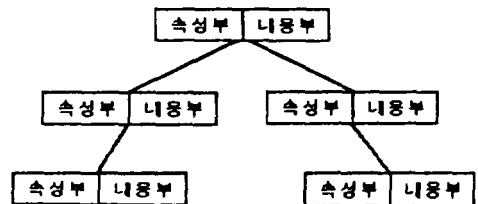
- 탐색 결과 목록(search result list): DFR 동작중의 하나인 탐색(search) 동작의 결과를 저장하는 DFR 객체이다. 그러므로 자주 탐색되는 DFR 객체인 경우 탐색 동작을 반복할 필요가 없다.

- 문서(document): 실제 문서의 데이터를 저장하는 곳이며, 사용자는 문서를 에디터로 작성한 후 문서 저장기의 최하위의 위치에 저장한다.

3. 확장된 디렉토리 시스템의 설계

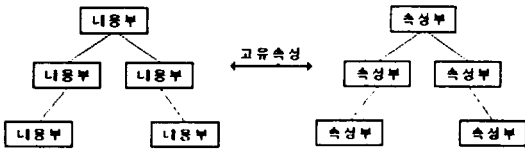
3.1 기본 개념

DFR 시스템의 문서 저장기에 화일링되는 DFR 객체는 속성부와 내용부로 구성되는데, 이를 화일 시스템에 화일링하는 경우 i) (그림 3)과 같이 속성부와 내용부를 하나의 화일에 저장하는 방법, ii) (그림 4)와 같이 내용부와 속성부를 별개의 화일에 저장하는 방법, iii) 화일 시스템을 확장하여 화일 시스템의 화일 속성을 수정하는 방법 세가지가 가능하나, DFR 시스템의 이식성을 높게 하기 위해서는, i), ii)의 방법이 바람직하다. 두가지 방법 중, DFR 시스템과 디렉토리



(그림 3) DFR 객체의 화일링 방법(1)
(Fig. 3) Filing method (1) of DFR Objects

시스템과의 연동을 위하여 i)의 방법보다는 내용부와 속성부를 별개의 화일에 저장하는 ii)의 방법이 적절하며, 디렉토리 시스템의 디렉토리 정보 트리(DIT: Directory Information Tree)를 확장하여 DFR 시스템의 속성부를 관리하도록 한다. DFR 시스템의 내용부와 속성부는 고유식별자(UPI)와 같은 고유속성을 이용하여 매핑되도록 하며, DFR 속성부는 문서 저장기 구조와 논리적으로 동일한 트리 구조를 이루도록 구성한다.

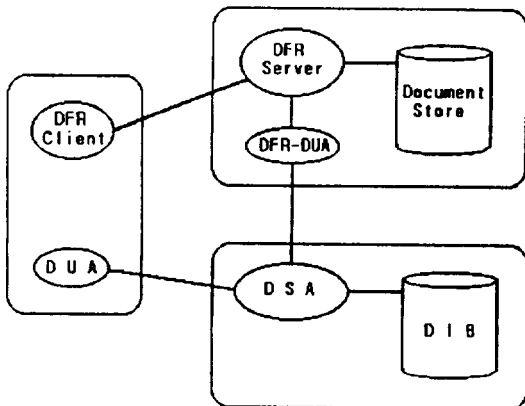


(그림 4) DFR 객체의 화일링 방법 (2)
(Fig. 4) Filing method (2) of DFR Objects

3.2 시스템 구성 모델

디렉토리 시스템과 DFR 시스템과의 연동을 위한 시스템 구성 모델은 (그림 5)와 같다.

(그림 5)와 같이 디렉토리 시스템과 DFR 시스템을 기존 사용 방식대로 각각의 클라이언트를 이용하여 개별적으로 사용할 수도 있으며, DFR 시스템의 탐색 동작 사용시 DFR 서버와 DSA간 인터페이스를 구성하는 DFR-DUA에 의하여 자동적으로 디렉토리 시스템을 이용할 수 있게 함으로써, DFR 사용시에는



(그림 5) 시스템 구성 모델
(Fig. 5) System configuration model

사용자에게 투명한(tranparent) 사용 환경을 제공하도록 한다.

기본적으로 DFR 탐색 동작 사용시 DFR 서버는 디렉토리 시스템을 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 디렉토리 내의 DFR 속성부와 DFR 문서 저장기가 동일한 트리 구조를 유지하도록 하여야 한다. 이를 위한 DFR 동작과 디렉토리 동작 사이의 매핑관계를 (표 1)에 나타내었다.

〈표 1〉 DFR, 디렉토리 동작 사이의 매핑관계
(Table 1) Mapping relationship between DFR and directory operations

기능	DFR 동작	디렉토리 동작	기능
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

3.3 확장된 디렉토리 스키마(schema)

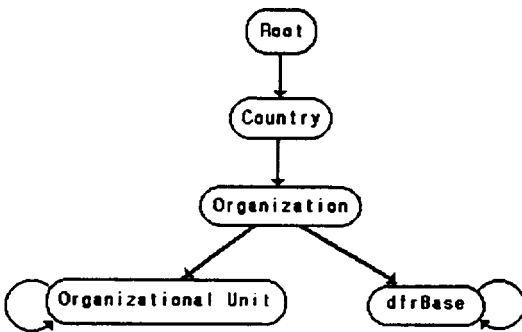
디렉토리 스키마는 DIT 구조, 객체 클래스, 속성 타입 및 구문(syntax)에 대한 정의 및 규칙을 표기하는 집합이며, 이는 부적합한 객체들의 생성, 객체에 해당되지 않은 속성 집합의 추가 및 갱신, 속성 타입에 어긋난 속성값들의 할당 등을 제어, 관리함으로써 효율적인 디렉토리를 유지하는데 중요한 기능을 담당한다.

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

에 기반을 두고 설계하도록 한다.

3.3.1 확장된 DIT 구조

기존 디렉토리 시스템의 DIT를 확장하여 DFR 속성을 관리하게 할 경우 확장된 DIT 구조와 구성에는 각각 (그림 6)과 (그림 7)에 보인 바와 같다.

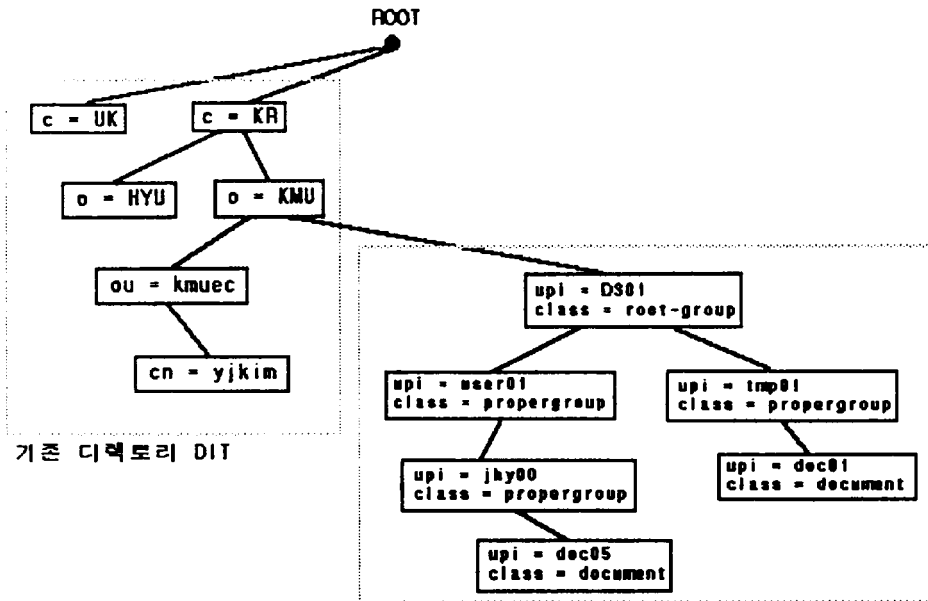


(그림 6) 확장된 DIT 구조
(Fig. 6) Extended DIT configuration

3.3.2 객체 클래스(object class)

(그림 6)에서와 같이 확장된 디렉토리 스키마에서는 X.521에 정의된 표준 객체 클래스인 country, organization, organizationalUnit 클래스를 그대로 이용하고, organization 클래스 아래 위치하는 dfrBase 객체 클래스를 다음과 같이 새로이 정의한다.

dfrBase OBJECT-CLASS
 SUBCLASS OF Top
 MUST CONTAIN {
 dfr-upi,
 dfr-object-class,
 dfr-title,
 dfr-pathame,
 version-name,
 create-date-and-time,
 created-by,
 dfr-access-list }
 MAY CONTAIN {



기존 디렉토리 DIT

확장된 DFR DIT

(그림 7) DIT 구성 모델
(Fig. 7) DIT configuration model

```
owner,
document-type,
subject }
```

3.3.3 속성

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

- dfr-upi: 고유식별자로써 DFR 서버에 의해 할당되며 DFR 객체를 고유하게 지정하는데 사용된다.

```
dfr-upi ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
        DfrUniquePermanentIdentifier }
DfrUniquePermanentIdentifier ::= OCTET STRING
```

- dfr-object-class: DFR 객체의 종류를 지정하는데 사용된다.

```
dfr-object-class ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
        DfrObjectClass }
DfrObjectClass ::= ENUMERATED {
    dfr-document (0),
    dfr-root-group (1),
    dfr-proper-group (2),
    dfr-reference (3),
    dfr-search-list (4)}
```

- dfr-title: DFR 객체의 이름을 나타내며, 사용자에게 의해 명시된다.

```
dfr-title ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax }
```

- dfr-pathname: 문서 저장기내에서의 DFR 객체의 위치를 나타내며, 루트 그룹으로부터 객체까지의 경로명이 지정된다.

```
dfr-pathname ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax }
```

- owner: DFR 객체의 소유자를 나타낸다.

```
owner ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX
        Person }
```

4. 구현 및 동작시험

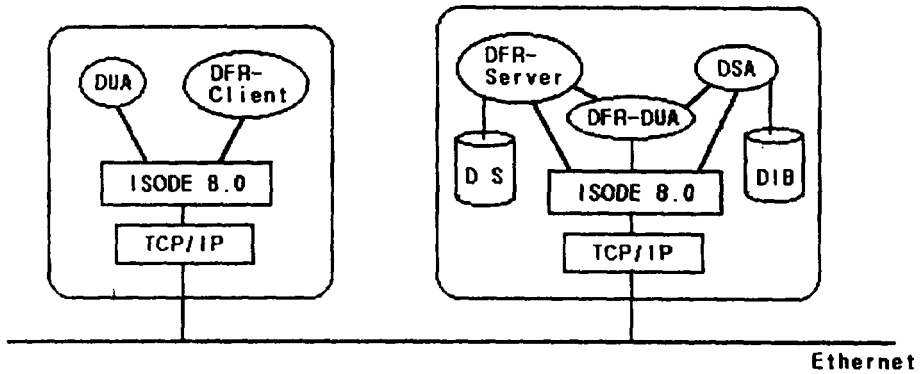
4.1 구현 환경

확장된 디렉토리 시스템의 구현 환경은 (그림 8)과 같이 Ethernet LAN으로 연결되어 있는 두대의 SUN 워크스테이션을 각각 클라이언트와 서버로서 사용하며, ISODE 개발툴[11]을 사용하도록 한다. DFR 시스템으로는 ISODE를 이용하여 개발된 HY-DFR 시스템[12]을 사용하고, ISODE 디렉토리 시스템인 QUIPU 8.0 소스 분석을 통해 확장 구현하여 HY-DFR 시스템과 연동시킨다. 그림에서는 DFR 서버와 DSA가 한 시스템에 존재하지만, DFR-DUA는 호스트 이름을 가지고 DSA에 접속을 하므로 DSA가 다른 시스템에 존재하여도 무방하다.

4.2 디렉토리와의 DFR 동작 사이의 연계 메카니즘

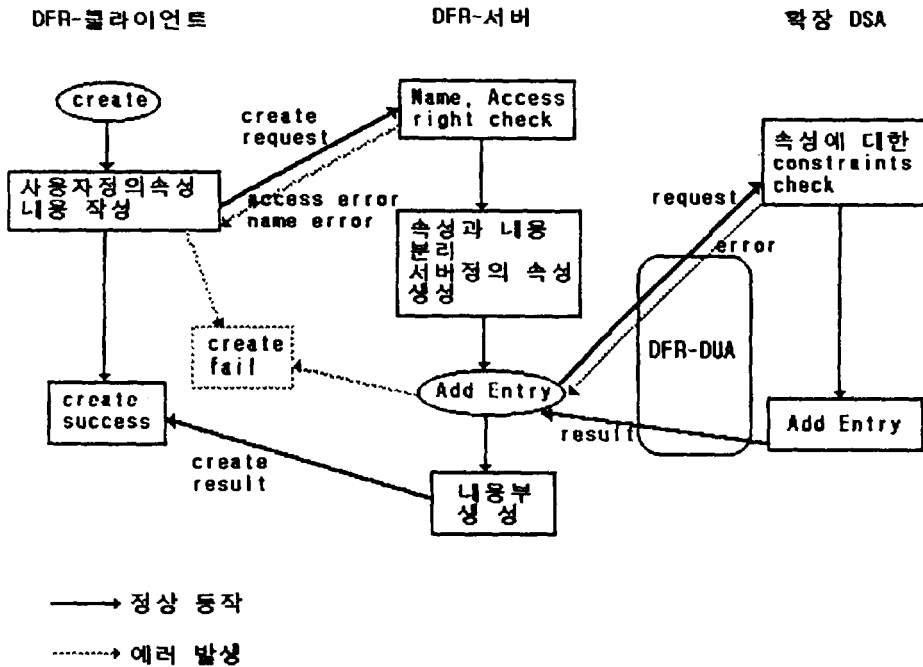
본 논문에서의 확장된 디렉토리 시스템은 기존 사용 방식대로 디렉토리 클라이언트인 DUA를 통하여, 기존의 사용자에게 대한 속성뿐 아니라 새로이 추가된 DFR 객체에 대한 속성도 검색할 수 있다. 또한 DFR 시스템 사용시에는 DFR 서버와 디렉토리 시스템 서버인 DSA간 인터페이스 역할을 하는 DFR-DUA에 의하여 자동적으로 디렉토리 시스템을 이용할 수 있게 함으로써, 사용자에게 투명한 사용 환경을 제공하게 된다. 따라서 DFR 사용시, DFR 클라이언트와 서버간 bind 동작이 수행되면 그와 동시에 DFR 서버는 DFR-DUA를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

확장된 디렉토리 시스템에서 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-EDB



DUA: Directory User Agent
 DSA: Directory System Agent
 DIB: Directory Information Base
 DFR: Document Filing & Retrieval
 DS: Document Store

(그림 8) 구현 환경
 (Fig. 8) Implementation environment



(그림 9) 생성 동작의 연계 메카니즘
 (Fig. 9) Association mechanism in create operation

가 수정된다. 본 절에서는 DFR 동작과 디렉토리 동작 사이의 주요 연계 메카니즘에 대하여 기술한다.

4.2.1 생성 동작

DFR 동작 중 생성 동작은 DFR 객체를 새로 생성하는 동작이므로, 디렉토리에 속성을 추가하는 AddEntry 동작과 연계가 이루어져야 하며, 연계 메카니즘은 (그림 9)와 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-DUA를 통하여 확장 디렉토리에 보내어 DFR-EDB에 엔트리를 추가하고 결과를 받으면 문서 저장기에 객체를 생성한 후 사용자에게 결과를 알려주게 된다.

4.2.2 탐색 동작

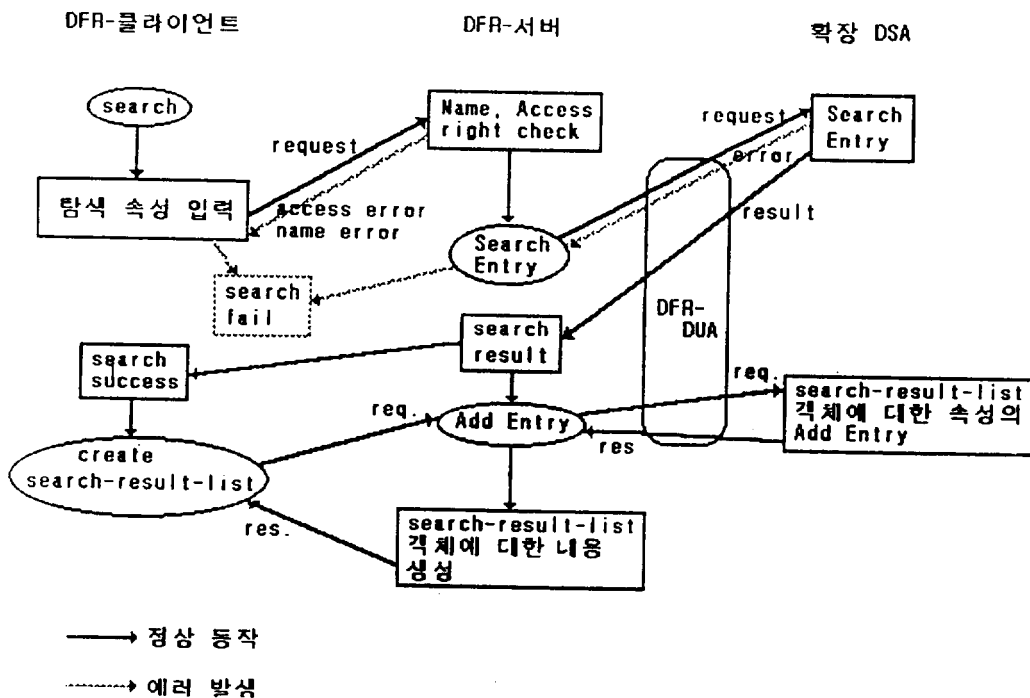
DFR의 탐색 동작은 디렉토리의 탐색 동작과 연계

가 이루어져야 하며, 연계 메카니즘은 (그림 10)과 같다. 사용자가 탐색하고자 하는 객체의 속성을 입력하여 탐색 동작을 요청하면, DFR 서버는 이를 DFR-DUA를 통하여 확장 디렉토리에 보내며, 그 결과를 사용자에게 알려준다. 이때, 사용자의 요청에 따라 그림에서와 같이 탐색 결과 목록(search result list)을 생성할 수도 있다.

4.3 동작시험 및 고찰

4.3.1 동작 시험

본 논문에서 구현한 확장된 디렉토리 시스템의 타당성을 위하여, 디렉토리 클라이언트인 DUA를 이용하여 DFR 객체를 탐색하는 방법과 DFR 시스템 사용시 탐색 동작을 이용하여 탐색하는 방법 두가지 동작 시험을 하였으며, 동일한 결과를 얻음으로써 디렉토리내의 DFR 속성에 대한 일관성과 사용방법에 대한 유연성을 확인하였다. 먼저 QUIPU의 DUA인 'dish'를 이용하여 기존에 디렉토리 시스템을 사용하



(그림 10) 탐색 동작의 연계 메카니즘
 (Fig. 10) Association mechanism in search operation

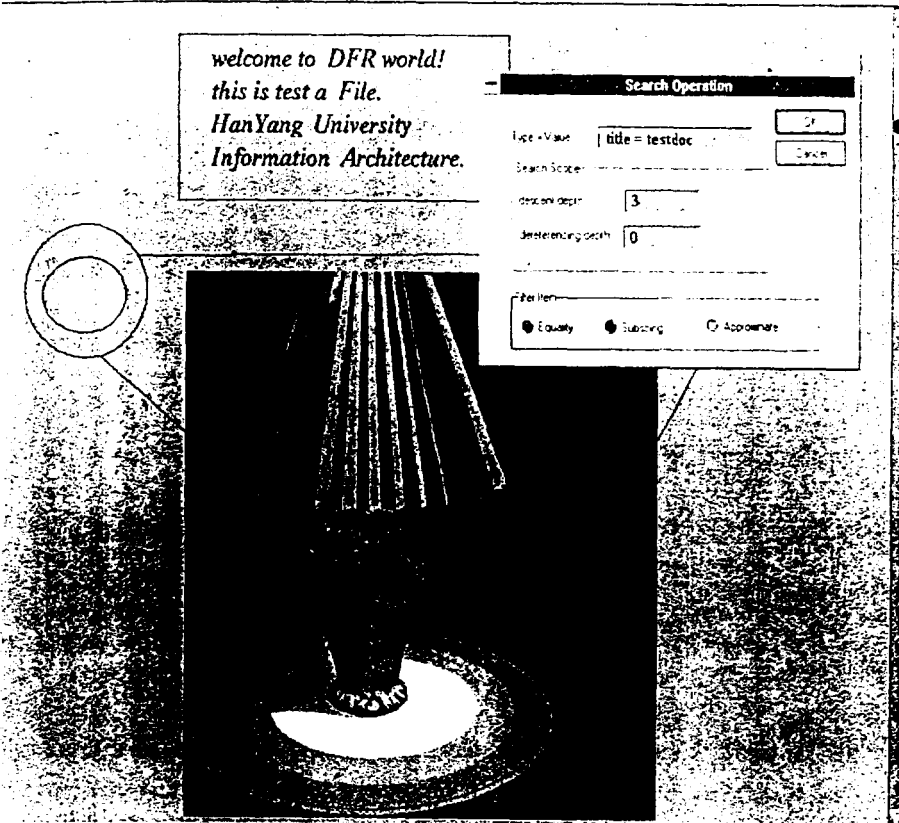
는 방법 그대로 DFR 객체를 탐색하였으며, DFR 시스템 사용시에는 (그림 11)과 같이 탐색 동작시 DFR-DUA에 의하여 자동적으로 확장 디렉토리를 이용하게 함으로써 투명성을 제공하였다. (그림 11)에서와 같이 DFR 탐색 동작을 사용하는 경우, 탐색하고자 하는 속성의 타입과 값을 입력하는 부분과 탐색 범위 (search scope), 필터를 지정할 수 있다. 탐색 범위에서 'descent depth'는 트리구조에서 탐색하고자 하는 DFR 일정 그룹의 깊이를 지정하며, 'dereferencing depth'는 DFR 객체가 참조되어 있는 경우, 어느 깊이만큼의 참조까지 탐색할 것인가를 지정하게 된다. 또한 필터 지정에 따라 정확한 매칭, 근사 매칭 기능을 선택할 수 있다.

4.3.2 고찰

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 다음과 같이 다양한 사무용 시스템의 객체 탐색에 디렉토리 시스템을 이용할 수 있는 확장 모델을 제시한다.

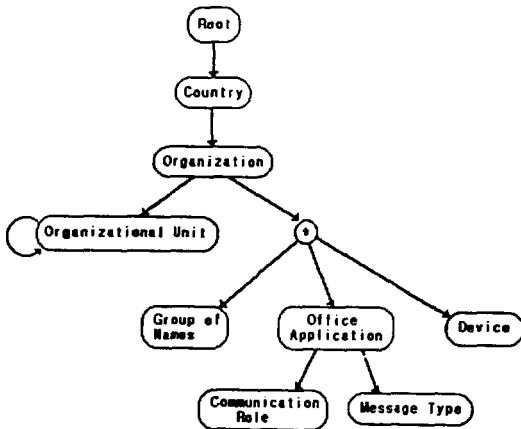
(그림 12)는 제안하는 DIT 구조를 나타낸다. 그림에서와 같이 그룹통신에 사용할 수 있는 GroupOfNames와 분산 환경에서 여러가지 공유장치를 지정할 수 있는 Device 객체 클래스를 정의함으로써 네트워크 사

create list read copy delete move modify reserve search drawing abandon



(그림 11) DFR 탐색 동작
(Fig. 11) DFR search operation

용자에 대한 정보외에 다양한 객체 관리에 디렉토리를 이용할 수 있도록 하고, 사무용용 시스템을 위한 OfficeApplication 클래스와 OA 클래스내에 문서나 메시지 타입, 기능역할을 나타내는 클래스를 정의함으로써 기존 디렉토리를 확장 사용할 수 있다.



(그림 12) 제안하는 DIT 구조
(Fig. 12) Suggested DIT structure

본 논문에서 구현한 확장 디렉토리 시스템은 기존 DUA 사용방법 그대로 사용할 수도 있으며, DFR 사용증에는 탐색 동작 사용만으로 디렉토리를 이용할 수 있다는 유연성이 제공된다. 그러나 대부분 문서 검색에는 문서 내용에 포함되는 키워드 검색을 하는 경우가 많은데, 본 논문에서의 시스템은 문서의 내용으로는 검색이 불가능하다는 단점이 있다. 이를 해결하기 위해서는 문서의 속성에 키워드를 추가하여 사용자가 키워드를 지정하게 하거나, 문서 내용을 검색할 수 있는 다른 방법을 제공해야 할 것이다.

5. 결 론

본 논문에서는 문서 화일링 및 검색(DFR: Document Filing and Retrieval) 시스템에서 DFR 객체의 속성(attribute)을 체계적으로 관리하여 DFR 객체의 효율적인 탐색(search) 기능을 제공하기 위한 확장된 디렉토리 시스템을 설계, 구현하였다. 이를 위하여 DFR 시스템과 디렉토리 시스템 상호연동을 위한 구성 모델 및 두 응용 시스템 동작들 사이의 연계 메카니즘,

확장된 디렉토리 스키마(schema), 확장된 디렉토리 트리 구조(DIT: Directory Information Tree)를 설계하고, 이를 ISODE 디렉토리 시스템인 QUIPU를 이용하여 구현하였다.

구현한 확장 디렉토리 시스템을 분산 환경하에서 QUIPU의 DUA인 dish를 이용하여 시험하고, DFR 탐색 동작을 통하여 두 시스템 사이의 연동 시험도 하였다. 또한 본 논문의 결과를 기반으로 다양한 응용 시스템의 객체 탐색을 위한 디렉토리 시스템 확장 모델을 제시하였다. 그러나 대부분 문서 검색에는 문서 내용에 포함되는 키워드 검색을 하는 경우가 많은데, 본 논문에서의 시스템은 문서의 내용으로는 검색이 불가능하다는 문제점이 있으므로, 향후 이를 보완하기 위하여 문서의 속성에 키워드를 추가하여 사용자가 키워드를 지정하게 하거나, 문서 내용을 검색할 수 있는 방법을 제공해야 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, Distributed 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 분산처리

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, 제31권 B편, 제2호, 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.



임 재 홍

1986년 서강대학교 전자공학과 졸업(공학사)
 1988년 한양대학교 대학원 전자공학과 졸업(공학석사)
 1995년 한양대학교 대학원 전자공학과 졸업(공학박사)
 1995년~현재 한국해양대학교 전자통신공학과 전임강사

관심분야: 그룹웨어, 분산처리, 초고속통신망



김 영 준

1986년 한양대학교 전자공학과 졸업(공학사)
 1990년 한양대학교 대학원 전자공학과 졸업(공학석사)
 1994년 한양대학교 대학원 전자공학과 박사과정 수료
 1996년 대전전문대학 비서행정과 전임강사

관심분야: 분산처리, 이동 통신