X.500 디렉토리를 이용한 관광 정보 스키마 설계

박 문 성[†] · 오 주 병[††] · 양 해 철[†††] · 이 용 준[††††] · 이 재 광[†††††]

요 약

X.500 디렉토리는 ISO(International Standards Organization)에서 제정한 컴퓨터 네트워크 모델인 OSI(Open Systems Interconnection) 참조 모델 7계층의 응용계층에 속하는 프로토콜로써, 정보통신 서비스에 필요한 정보를 데이터베이스화하여 효율적으로 관리하고, 사용자가 편리하게 접근할 수 있는 기능을 제공하는 서비스다. 관광 정보는 여러 위치들로 분산되어 있으며, 분산된 정보의 액세스는 X.500 디렉토리 서비스와 유사하므로 본 논문에서는 X.500 디렉토리 서비스를 이용하여 관광 정보 시스템과 디렉토리 스키마를 설계하였다. 관광 정보 엔트리를 정의한 후, 관광 정보인 산, 바다, 박물관, 호텔 등의 새로운 객체 클래스와 지역적인 정보를 서비스하기 위한 디렉토리 정보 트리를 제안하였으며, test-bed를 구성하여 설계한 디렉토리 스키마를 검증하였다.

Tourist Information Schema Design Using X.500 Directory

Moon Sung Park[†] · Joo Byung Oh[††] · Hae Chul Yang[†††] · Ryong Joon Lee[††††] · Jae Kwang Lee[†††††]

ABSTRACT

X.500 Directory is an application level protocol in the Open Systems Interconnection(OSI) 7 layer's reference model adopted by the International Standards Organization(ISO). It manages effectively needed information for the communication service, and is a service to support the functions that the user can approach conveniently. Tourist information is distribution on several locations. The access of distributed information is similar to X.500 directory service. Therefore, we design of Tourist Information System(TIS) and directory schema using X.500 directory services. To define tourist information, we also propose Directory Information Tree(DIT) of locality architecture and several new object classes, such as museum, mountain, sea, hotel, etc.

1. 서 론

오늘날 정보 통신망의 규모가 커지고 전세계적인 정보 통신망(Internet)이 구성되어 통신망에 연결된 정보 기기(예. 호스트 컴퓨터, 통신 장비)와 가입자들이 크게 늘어나고, 이들의 통신망 주소(network address) 등과 같은 정보 통신 서비스에 필요한 모든 정보들이 복잡하고, 대량화됨에 따라 이를 체계적이고, 효율적으로 저장 및 관리할 수 있는 새로운 서비스의 필요성이 요구되었다.

디렉토리란 이러한 요구에 따라 정보 통신 서비스에 필요한 모든 정보를 데이터베이스화하여 이를 효율적으로 관리하고, 사용자가 온라인으로 편리하게 사용할 수 있는 기능을 제공하는 서비스이다[1, 2].

디렉토리의 주요 기능은 이름과 주소(name-to-ad-

† 정 회 원:한국전자통신연구소 정보공학연구실
†† 정 회 원:한국전자통신연구소 정보공학연구실
††† 정 회 원:대전전문대학 전자계산소
†††† 정 회 원:한국전자통신연구소 정보자동화연구실
††††† 정 회 원:한남대학교 전자계산공학과 조교수

논문접수:1996년 3월 26일, 심사완료:1996년 5월 1일

dress)의 매핑(mapping)으로 전화 번호부의 상대방 이름, 직업 등을 키(key)로 하여 전화 번호를 찾기 위해 사용되는 것과 유사하게, E-mail에서 상대방 주소를 모르더라도 이름 등의 친숙한 키를 이용하여 검색한 주소로 E-mail을 보내지도록 한다. 또한 디렉토리는 다양한 데이터 유형(type)을 지원하며, 강력한 검색 기능을 제공하므로 여러 응용 분야(예. 차세대 지능망 서비스를 위한 가입자 데이터베이스)에 이용될 수 있다. 디렉토리는 ISO에서 제정한 OSI참조 모델의 7계층 중 가장 상위 계층인 응용 계층(Application Layer)에 속하는 프로토콜로써 MHS, FTAM(File Transfer Access and Management)같은 다른 응용 계층 프로토콜들이 통신 시에 필요한 서비스 정보를 디렉토리 시스템에 질의하여 결과를 얻을 수 있다.

본 논문에서는 이 디렉토리에 지역적 정보(예. 국가, 또 혹은 직할시, 군 및 시, 관광 이름 등에 대한 정보)인 관광 정보를 추가하여, 디렉토리 서비스를 제공할 수 있도록 디렉토리 시스템을 구성함으로써, 조직적인 정보뿐만 아니라 지역적인 정보에 의한 서비스도 가능하게 하여, 개방형 정보 통신 응용 서비스로서의 확장이 가능함을 보였다. 2장에서 디렉토리의 개념을 설명하였고, 3장에서는 조직적 정보에 지역적 정보를 포함하여 서비스할 수 있도록 디렉토리 시스템을 설계하였으며, 4장에서는 스키마 설계 방법에 의한 관광 정보의 스키마를 설계하고, 지역적인 정보를 검색할 수 있는 DIT 구조를 제안하였다. 그리고 5장에서는 이 관광 정보 시스템의 적용을 위한 test-bed 구성 및 시험을 하였으며, 6장에서는 향후 연구 방향을 제시하였다.

2. X.500 디렉토리의 개요

이 장에서는 관광 정보 서비스를 제공하기 위해 필요한 기술을 정리하여 관광 정보 서비스를 하기 위한 환경에 대하여 기술하였다. 이 절의 주요 내용은 디렉토리 정보를 저장하기 위한 저장소와 디렉토리의 계층 구조, 서비스를 액세스하기 위한 추상 서비스 동작 그리고 디렉토리 시스템간의 상호 운영성에 대한 분산 동작 등에 관한 것이다.

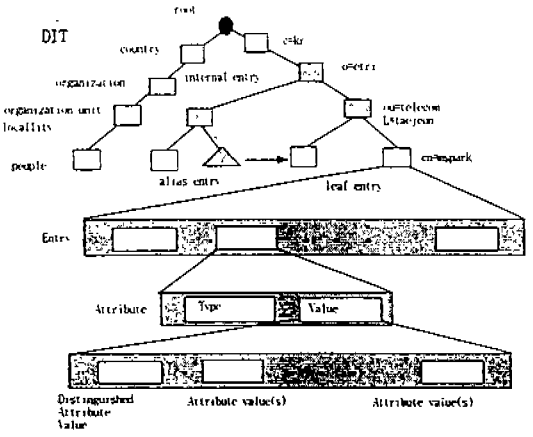
2.1 디렉토리 정보 베이스

디렉토리는 객체지향 모델링(object-oriented modelling) 개념을 도입하여, 실세계의 모든 정보를 객체로 모델링 하여 표현한다. 디렉토리 객체는 사람, 조직, 서비스 등이 되며, 객체지향 프로그래밍에서의 객체 개념과는 달리 데이터와 함수가 캡슐화(encapsulation)된 객체가 아니라 단지 데이터만을 표현한다. 한 디렉토리 객체는 하나의 엔트리(Entry)에 저장되며, 엔트리의 전체 집합을 디렉토리 정보 베이스(Directory Information Base:DIB)라고 한다. 엔트리는 속성의 집합으로 구성되며, 디렉토리 표준에서는 국가명, 지역명, 조직명, 우편번호, 직책, 전화번호 등 약 60여개의 속성 유형을 정의하고 있으며, 필요에 따라 사용자가 새로운 속성 유형을 정의할 수 있다. 그러나 다른 사용자가 같은 이름의 속성 유형을 정의하는 것을 방지하기 위해 각 속성 유형에는 고유의 객체 식별자(Object IDentifier:OID)는 연속성 정수들로 구성된다. 일반적으로 디렉토리에서는 속성을 구분하기 위하여 속성의 이름을 사용하지 않고, OID를 사용한다. 각 속성의 유형은 속성이 저장될 속성 유형을 나타내는 속성 구문(attribute syntax)과 속성 값을 서로 비교하는 매칭 규칙(matching rule)을 가진다. 디렉토리 표준에서는 각 속성 유형에 대한 속성 구분과 매칭 규칙을 ASN.1(Abstract Syntax Notation One)으로 상세히 명시하고 있다[3, 4].

2.2 디렉토리 정보 트리

DIB내의 모든 엔트리는 UNIX 파일 시스템과 유사하게 계층적으로 배열되어 트리 구조를 디렉토리 정보 트리(Directory Information Tree:DIT)라고 부른다. DIT 내의 모든 엔트리는 특정 속성들로 구성된 상대 고유 이름(Relative Distinguished Name:RDN)이라는 이름을 가지는데, RDN은 파일시스템에서 한 디렉토리 밑의 파일을 구분하기 위해 이름이 사용되는 것과 같이 같은 상위 엔트리를 가진 하위 엔트리를 구분하기 위해 사용된다. DIT내의 각 엔트리는 루트(root)에서부터 엔트리의 위치까지 연속된 RDN의 집합으로 고유(unique)하게 구별되는데, 이 RDN의 집합을 고유 이름(Distinguished Name:DN)이라고 부르며, 파일시스템에서의 전체 경로(path) 이름과 유사하다. RDN과 DN은 디렉토리 정보를 액세스하기 위한 키로서 사용되므로 고유해야 하는 것은 물론 기

억하고, 디렉토리 사용자가 이해하기 쉬워야 하므로 별도의 명명 기관(Naming Authority)에 의해 할당되고, 관리되는 것이 바람직하다(예. 박문성의 DN은 {c=한국, o=전자통신연구소, ou=정보시스템연구부, cn=박문성}이 되며, c는 countryName, o는 organizationName, ou는 organizationUnit Name 그리고 cn은 commonName이 된다)[4].



(그림 2) DIT와 엔트리의 속성
(Fig. 2) The Attribute of Entry and DIT

2.3 객체 클래스

디렉토리 엔트리는 객체에 대한 정보를 저장하는데, 동일한 속성들로 구성된 객체의 집단을 객체 클래스라고 하며, 각 엔트리는 특정 클래스에 속해 있다. 각 클래스의 엔트리는 필수(mandatory) 및 선택(optional) 속성들로 구성되며, 클래스마다 서로 다른 필수, 선택 속성을 갖는다. 이 객체 클래스는 상속(inheritance)의 개념을 가지며, 객체 클래스의 사양을 정의하는데 있어서 상위 클래스는 필수 및 선택 속성을 정의하고, 하위 클래스는 그것의 상위 클래스의 필수 및 선택 속성을 상속받는다[4, 7, 8, 9].

2.4 디렉토리 추상 서비스

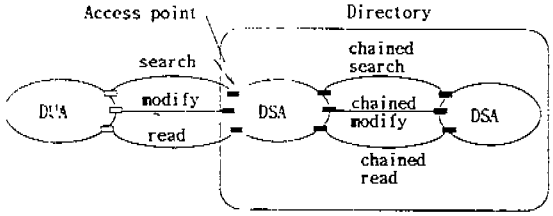
디렉토리에 의해 제공되는 서비스 동작은 크게 디렉토리 정보를 찾아내는 질의와 디렉토리 정보의 구조 및 내용을 수정하는 변경으로 분류된다[7, 10].

- 읽기 동작(Read operation): 읽기 요청은 특정 엔트리를 목표로 하며, 엔트리 속성의 전부 또는 일부 값을 디렉토리에서 읽기 위한 서비스 동작이다.
- 비교 동작(Compare operation): 이것은 명시된 엔트리의 어떤 속성 유형에 대한 값을 사용자가 원하는 값과 비교하는 동작으로서, true나 false로 비교 결과를 응답하는 서비스 동작이다.
- 탐색 동작(Search operation): 이것은 지정된 객체에서부터 하위 엔트리들에 대하여 제시된 조건에 맞는 엔트리들을 찾아서 선택된 정보를 출력해 주는 서비스 동작이다.
- 열람 동작(List operation): 이것은 지정된 객체의 모든 하위 엔트리들을 찾아서 출력해 주는 서비스 동작이다.
- 엔트리 추가 동작(AddEntry operation): 이것은 DIT에 새로운 엔트리인 leafEntry(object Entry/aliasEntry)를 추가하는 동작이다.
- 엔트리 제거 동작(RemoveEntry operation): 이것은 DIT에서 지정된 엔트리(leafEntry)를 삭제하는 서비스 동작이다.
- 엔트리 수정 동작(ModifyEntry operation): 이것은 현재 DIT에 존재하는 하나의 엔트리를 대상으로 새로운 속성 유형을 추가 또는 지정된 속성을 삭제하거나, 새로운 속성값을 추가 또는 지정된 속성값을 삭제하는 서비스 동작이다.
- RDN 수정 동작(ModifyRDN operation): 이것은 현재 DIT에 존재하는 leafEntry(object Entry/aliasEntry)를 대상으로 RDN을 변경시켜 주는 동작이다.

2.5 디렉토리 분산 동작

디렉토리는 클라이언트/서버 모델을 기본으로 하고 있다. 사용자(클라이언트)들은 하나의 디렉토리 사용자 처리기(Directory User Agent: DUA)를 통해 디렉토리에 접근하게 되며, 이것을 접근점(access point)라 한다. DUA는 사용자 측면에서 서비스 처리를 위하여 디렉토리 오퍼레이션으로 액세스하는 방법을 제공한다. 이 디렉토리는 DUA에게 하나의 단일 논리 실체처럼 보인다 할 지라도 이것은 많은 물리적 디렉토리 시스템 처리기(Directory System Agent: DSA)로 구성된 거대한 데이터베이스이다. 디렉토리 액세스 프로토콜(Directory Access Protocol: DAP)은 DUA와 DSA간 인터페이스를 위해 사용되며, 시스템이 제공

하는 서비스와 디렉토리간의 통신 표준이다. 디렉토리 시스템 프로토콜(Directory System Protocol : DSP)은 DSA와 DSA간 인터페이스에 사용되며, 분산된 DSA들 간의 상호 통신을 위한 표준이다. 이 모델에서는 DUA로부터 디렉토리 추상 서비스(Read, Search, Modify)를 요구(Request)받은 DSA는 해당 정보를 가지고 있는 다른 DSA에 연결하여 DSA 추상 서비스(ChainedRead, ChainedSearch, Chained Modify)를 요청하고, 그 결과를 전달받아 디렉토리 추상 서비스를 처리한 후, DUA에게 응답(Response)한다[5, 6, 11].

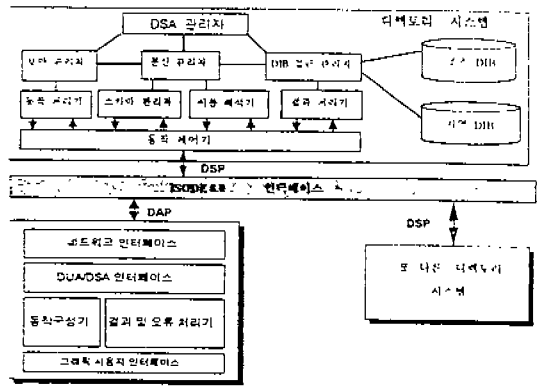


(그림 3) 분산 디렉토리 모델의 객체
(Fig. 3) Objects of the Distributed Directory Model

3. 디렉토리 시스템 설계

디렉토리는 통신망 상의 여러 사이트(site)에 분산되어 동작하는 반면 사용자에게는 집중화된 단일 시스템으로 보이므로 일종의 분산 데이터베이스 시스템으로 볼 수 있으며, 다음과 같은 특성을 갖고 있다[3, 5].

- 디렉토리는 변경(update)보다는 질의(query)의 빈도가 훨씬 높은 전기 통신 서비스에 적합하도록 설계되었다. 디렉토리 표준은 질의에 대하여 다양한 기능을 제공하나 변경에 대해서는 기본적인 기능만을 명시하고 있다.
- 디렉토리에서는 비일관성(inconsistency)이 어느정도 허가된다. 예를 들어 성능과 신뢰성을 위해 디렉토리 정보는 여러 사이트에 복제(replication)될 수 있으나, 원(original) 정보에 대한 변경이 발생될 경우에는 즉각적인 변경이 요구되지 않는다. 따라서 다른 사이트에 저장된 같은 정보를 동시에 검색을 시도하면 검색 결과가 서로 다를 수 있다.



(그림 4) 디렉토리 시스템 모델
(Fig. 4) Directory System Model

- 디렉토리에서는 잠금(locking)과 같은 동시성 제어 기능을 제공하지 않는다. 두 사용자가 동시에 같은 정보를 변경했을 경우에는 정보에 대한 일관성이 보장되지 않는다.

이러한 특성 때문에 디렉토리는 트랜잭션 처리보다는 정보의 비일관성을 용인할 수 있는 정보 검색등과 같은 응용 분야에 적합하다. 이와 같은 기능을 고려하여 지역적인 정보를 갖는 관광 정보 서비스를 디렉토리 서비스에 적용하기 위한 방법을 제시하고, 디렉토리 시스템을 설계하였으며, 분산 동작 수행이 용이하도록 다음과 같은 기능 모듈로 설계하였다.

- 동작 제어기 : DUA로부터 동작 요청을 받아 해당 기능 블록으로 전달하고, 제어하는 역할을 수행하며, 동작 수행 후에 얻은 결과나 다른 디렉토리 시스템으로부터 받은 결과를 요청한 DUA로 반환하는 기능을 수행한다.
- 이름 해석기 : 스키마 관리자와 분산 관리자를 통해 얻은 지식 정보 테이블에 있는 디렉토리 시스템의 문맥 접두사와 엔트리의 이름을 비교하여, 그 엔트리가 국부 디렉토리 시스템에 있는지 다른 디렉토리 시스템에 있는지를 결정한다. 국부 디렉토리 시스템에 의해 접근할 수 있는 엔트리이면, 그 엔트리에 대한 DIB 정보(객체 부류의 키)를 반환하고, 다른 디렉토리 시스템에 있는 엔트리이면 디렉토리 시스템의 시스템의 시스템 이름을 반환한다.
- 동작 처리기 : 스키마 관리자와 이름 해석기에서 얻은 객체 부류 테이블과 키를 이용하여, 요청된 동

서에 따라 객체 분류 테이블로부터 정보를 얻는다. 처리되는 동작은 (chained)compare, (chained)search, (chained)modifyEntry, (chained)addEntry, (chained)removeEntry 등이 있다. 위의 동작 중 변경 동작을 수행할 때는 다른 사용자에 의해 변경 동작에서 조작되는 테이블의 접근을 막기 위해 그 테이블의 잠금 뒤에 동작을 처리한다.

- 스키마 관리자:DIB의 구조를 정의한 디렉토리 스키마에 대한 정보를 저장, 관리하는 역할을 수행한다. 내부에 존재하는 별도의 스키마 저장소(schema repository)에 디렉토리 스키마 정보를 저장하여 클래스와 속성의 생성, 추가, 변경 및 DIT 구조의 변경이 발생할 경우 관련된 스키마 정보를 변경한다. 스키마 정보 변경 시 ASN.1으로 정의된 변경 정보를 DIB 접근 관리자에 전달한다.
- 결과 처리기:각 기능 모듈에서 기능 처리 오류중 오류가 발생한 경우에, 그 오류의 종류에 따라 오류를 설정하거나, 동작 처리 후 얻은 결과를 DUA로 반환한다. 이름 해석기나 동작 처리기에서 다른 디렉토리 시스템에 있는 엔트리에 대한 정보를 얻어야 할 경우에는 다른 디렉토리 시스템으로 동작(ChainedX)을 요청하여 얻은 결과를 현 디렉토리 시스템에서 얻은 결과와 결합한다.
- DSA 관리자:독립적으로 동작하는 프로세서로써 임의의 시스템에서 디렉토리 시스템을 새로 구성할 때 디렉토리 시스템이 필요로 하는 지식 정보를 이용하여 지식 정보 파일을 구성한다. 이 때 필요한 지식 정보는 문맥 접두사, 상위 참조, 하위 참조, 불특정 하위 참조, 디렉토리 시스템 이름 등이 있다. 이 기능은 디렉토리 시스템을 효율적으로 관리하기 위해 부가적으로 필요한 기능을 수행하는 프로그램들의 집합이다. 디렉토리 시스템의 시스템 부팅, 모니터링 프로그램과 백업, Export/Import 프로그램 등으로 구성되어 있다.
- 보안 관리자:DIB 정보를 불법적인 액세스로부터 보호하기 위해 공개키 암호화 기법을 이용한 접근 제어를 담당한다. 디렉토리 이용자의 비밀키를 관리하고, 이용자에게 키를 분배하는 키 관리 기능을 가지며, DIB의 데이터를 안전하게 보호한다.
- DIB 접근 관리자:대량의 데이터를 효율적으로 관리할 수 있는 DBMS로서 DIB내의 디렉토리 데이터를 물리적으로 저장, 처리하는 역할을 담당하며 실시간 데이터 처리, 데이터 복구, 동시성 제어, 무결성 검사 기능 등을 수행하며, 조직적 DIB와 지역적 DIB로의 접근을 관리한다.
- 분산 관리자:디렉토리 시스템이 분산된 여러 다른 디렉토리 시스템과 상호 정보 교환을 통해 데이터를 처리할 수 있는 역할을 수행한다. 분산 디렉토리 시스템에 대한 지식을 저장, 관리하는 지식 관리 기능과 데이터 복제 기능을 수행한다.

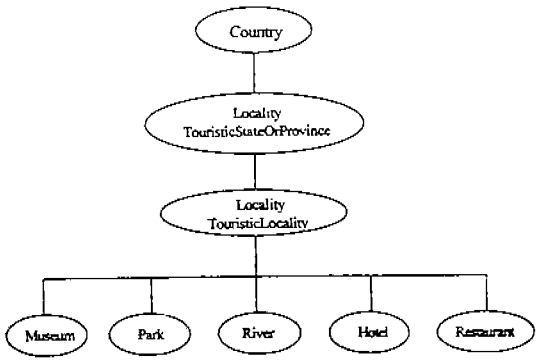
4. 관광 정보 스키마 설계

디렉토리는 일종의 데이터베이스 시스템이므로 상용 DBMS를 사용하여 DB 응용 프래그램을 개발할 경우에 DB 스키마를 설계해야 하는 것과 마찬가지로 디렉토리 응용을 개발하기 위해서는 우선적으로 디렉토리 스키마가 설계되어야 한다. 디렉토리 스키마의 설계 단계는 다음과 같으며, ASN.1을 사용하여 스키마를 정의한다[3, 4].

1) DIB 구성에 필요한 객체 클래스와 속성 유형(필수 속성, 선택 속성)을 정의한다. 필요한 클래스와 속성 유형이 표준에 정의되어 있는 경우 표준을 사용하고, 없을 경우에는 새로운 클래스와 속성 유형을 임의로 정의한다.
2) DIT 엔트리의 이름 결합(name binding)을 정의한다. 즉, DIT 엔트리가 속하는 클래스와 DIT 엔트리의 RDN을 구성하는 속성들을 결정한다.
3) DIT를 구성하는 엔트리들 간의 상하 계층 관계를 정의한다.

또한 디렉토리에는 사용자 정보외에도 디렉토리 관리에 필요한 관리 정보가 저장되어야 하므로 사용자 정보를 정의한 디렉토리 스키마뿐만 아니라 관리 정보를 정의한 디렉토리 시스템 스키마의 설계도 역시 필요하다. 본 논문에서 제안한 관광 정보 스키마는 X.500 디렉토리 스키마에 근간을 둔다. 그러므로 X.500 디렉토리에서 제안된 객체 클래스들의 일부 클래스 및 속성들은 관광 엔트리들의 기본 클래스와 속성으로 이용된다. 이 외에 관광 정보를 위해 산, 바다, 강, 공원, 호텔, 레스토랑 등 새로운 객체 클래스들과 이 클래스들에 포함되는 속성을 정의할 수 있다. 본 논문에서는 관광 정보의 디렉토리 스키마 속성들 중

에서 일부분만을 기술한 것이다.



(그림 5) 관광 정보 객체 클래스 DIT
(Fig. 5) DIT of Tourist Information Object Class

4.1 클래스

관광 정보 스키마를 구성하는 전체 클래스 구조를 DIT의 상하 관계에 의거하여 나타낸 것이며, 관광 정보 객체 클래스 DIT는 (그림 5)와 같이 정의하였다.

4.1.1 기본 객체 클래스

관광 정보는 X.521에서 선정된 객체 클래스 중 country, locality와 같은 기본 객체 클래스를 이용한다. 다음은 X.521에서 정의된 ASN.1으로 표기된 country의 객체 클래스를 나타낸다[1, 13, 14].

```
country OBJECT-CLASS ::= {
  MUST CONTAIN {
        countryName}
  MAY CONTAIN {
        description,
        searchGuide}
  {objectClass 2}
```

하나의 객체 클래스는 MUST 속성과 MAY 속성으로 분류될 수 있는데 MUST는 필수(mandatory) 사항이고, MAY는 선택(optional) 사항이다.

countryName은 국가 이름을 나타내고, description은 국가에 대한 관광 정보 내용을 서술하는 선택적인 속성이다. 또한 선택 속성은 필요에 의해 추가할 수 있다.

4.1.2 관광 정보 객체 클래스

1) Touristic State or Province

touristicStateOrProvince 객체 클래스는 DIT에서 하위에 각 지역에 대한 관광 정보를 가지고 있는 시, 도를 나타내는 엔트리이다.

```
touristicStateOrProvince OBJECT-CLASS
SUBCLASS OF top
MUST CONTAIN {
        stateOrProvince }
MAY CONTAIN {
        description }
```

2) Touristic Locality

touristicLocality 객체 클래스는 유용한 관광 정보를 포함하고 있는 지역인 시, 군을 나타내는 엔트리들을 정의하는데 사용된다.

```
touristicLocality OBJECT-CLASS
SUBCLASS OF top
MUST CONTAIN {
        localityName }
MAY CONTAIN {
        description }
```

3) Museum

museum 객체 클래스는 박문관에 포함된 엔트리들을 정의한다(예. 부여 박물관).

```
museum OBJECT-CLASS
SUBCLASS OF top
MUST CONTAIN {
        commonName.
        completeLocation.
        description }
MAY CONTAIN {
        yearOfEstablishment,
        scale,
        collection,
        close,
        inspectionTimes ... }
```

4.2 속 성

관광 정보 스키마는 표준안에서 정의된 기본 속성을 사용하며, 기본 속성에 포함되지 않은 속성을 정

의하여 사용한다.

4.2.1 기본 속성

X.520에서 정의한 속성 중 다음 속성을 기본 속성으로 사용한다.

- contryName
- stateOrProvince
- localityName
- description
- commonName

contryName을 ASN.1으로 표현하면 다음과 같다[7, 13].

```
conrtryName ATTRIBUTE ::= {
        WITH ATTRIBUTE SYNTAX
        PrintableString(SIZE(2))
                -IS 3166 codes only
        MATCHES FOR EQUALITY
        SINGLE VALUE
                ::= {attributeType 6}
```

4.2.2 관광 정보를 위한 속성

1) yearOfEstablishment

yearOfEstablishment 속성은 설립 년도를 나타낸다.

```
yearOfEstablishment ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX
        numericStringSyntax
        ID      {attributeType ..}
```

2) scale

scale 속성은 면적을 나타낸다.

```
scale ATTRIBUTE ::=
        WITH ATTRIBUTE-SYNTAX
        integerSyntax
        ID      {attributeType ..}
```

3) collection

collection 속성은 박물관에 소장품을 나타낸다.

```
collection ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
        ID      {attributeType ..}
```

4) close

close 속성은 휴관일을 나타낸다.

```
close ATTRIBUTE ::=
WITH ATTRIBUTE-SYNTAX
        intergerSyntax
        ID      {attributeType ..}
```

5) inspectionTime

inspectionTime 속성은 관람 가능 시간을 나타낸다.

```
inspectionTime ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX
        numericStringSyntax
        ID      {attributeType ..}
```

4.3 이름 결합

디렉토리 이름은 모든 객체들로부터 특정 객체를 추출하는 구조체이다. 이름은 한 개의 객체만을 지칭하도록 모호하지 않아야 한다. 이름 결합을 위해 ASN.1으로 다음과 같이 정의하였다[3, 4].

◦ countryName 이름 결합

```
countryNameBinding NAME-BINDING ::= {
        NAMES           country
        WITH ATTRIBUTES {countryName}
        ID              {nameBinding 1}}
```

◦ touristicStateOrProvince 이름 결합

```
touristicStateOrProvinceNameBinding ::= {
        NAMES           touristicStateOrProvice
        WITH ATTRIBUTES
                {touristicStateOrProvince}
        ID              {nameBinding ..}}
```

◦ touristicLocality 이름 결합

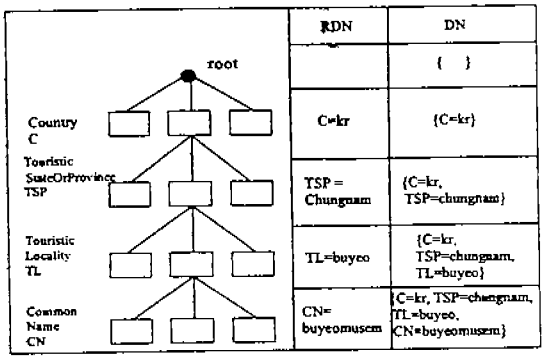
```
touristicLocalityNameBinding ::=
        NAMES           touristicLocality
        WITH ATTRIBUTES
                {touristicLocality}
        ID              {nameBinding ..}}
```

◦ museum 이름 결합

```
museumNameBinding ::= {
```

NAMES museum
WITH ATTRIBUTES {museum}
ID {nameBinding ..}}

4.4 상하 계층 관계의 정의



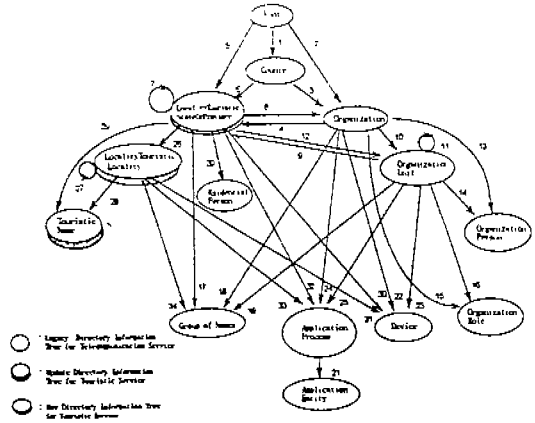
RDN	DN
	{ }
C=kr	{C=kr}
TSP = Chungnam	{C=kr, TSP=chungnam}
TL=buyeo	{C=kr, TSP=chungnam, TL=buyeo}
CN= buyeomusem	{C=kr, TSP=chungnam, TL=buyeo, CN=buyeomusem}

(그림 6) 관광 정보의 고유 이름과 상대 이름 비교
(Fig. 6) DN and RDN for Tourist Information

각 엔트리는 유일한 RDN을 갖는다. RDN은 엔트리의 고유값에 관하여 참 값을 갖는 속성값의 집합으로 이루어지며, 이 집합은 엔트리에 있는 각 고유값에 대하여 정확하게 한 개의 지정만 갖는다. 주어진 객체의 고유이름은 객체와 그 객체의 모든 상위엔트리(내림차순)를 표현하는 엔트리의 RDN 순서로 식별한다. 객체와 객체 엔트리 사이가 1:1 대응이므로 객체의 고유 이름도 객체 엔트리를 식별할 수 있다. (그림 6)은 RDN과 DN을 통하여 필요한 관광 정보를 결정하는 예를 보인 것이다.

4.5 관광 정보를 위한 DIT 구조 제안

본 절에서는 기존의 DIT에 지역적 정보를 처리할 수 있도록 새로운 DIT를 추가하였다. 기존 DIT는 조직적인 정보의 서비스를 위한 형태로 구성되어 있으며, 지역적인 정보는 거주자의 정보만을 포함하고 있다. 이러한 구조에 지역적 DIT 구조를 추가하기 위하여 Country의 Organization과 같은 객체 클래스로서 존재하는 Locality에 LocalityTourist, StateOrProvince로 수정하고, 하위 계층에는 LoaclityTouristLocality와 TouristName이라 정의하여 지역적 정보의 서비스도 가능한 DIT구조를 제안하였다.



(그림 7) 관광 정보를 위한 DIT 구조 제안
(Fig. 7) Suggest DIT Structure for Tourist Information

5. Test-bed 구성 및 시험

앞에서는 설계한 디렉토리 스키마의 검증을 위하여 스키마를 실제로 구현하여, 관광 정보 디렉토리를 만들고, 이를 정보 통신망으로 상호 연결하여, 분산된 디렉토리로 확장하여 디렉토리간의 상호 운용성(interoperability)을 시험할 수 있는 test-bed를 구축해 보았다. 이를 위한 시험 환경은 다음과 같다.

- 시험 환경 구성 : 소프트웨어로 가장 널리 사용되고 있는 QUIPU가 있으며, QUIPU는 소스(source)가 공개된 소프트웨어로 ISODE(ISO Development Equipment)의 일부분을 제공하며, 많은 시범 프로젝트에 의해 만들어진 것이다. QUIPU DSA는 UNIX 파일 시스템을 이용하여 DIB를 저장하며, QUIPU DUA는 API(Application Program Interface)를 제공한다.
- DSA : UNIX 시스템의 TCP 프로토콜 상에서 ISODE 8.0 인터페이스와 QUIPU를 설치하여, DSA를 구성하고, QUIPU의 EDB(Entry Data Block)들의 계층적인 구조에 관광 정보의 스키마와 관광 정보를 정의하였다.
- DUA : TCP/IP환경과 윈도우 소켓에서 동작하는 LDAP(Light-weight Directory Access Protocol) 사용자 인터페이스를 구성하였다. 또한 DSA(QUIPU 서버)를 액세스하기 위해 드라이버를 visual C++ 언어로 개발하고, 클라이언트의 동작은 디렉토리

의 DUA와 같은 기능을 수행하도록 구성하였다.

DUA와 DSA의 연결은 LDAP을 사용하여 디렉토리 결합을 수행하고, 디렉토리 추상 서비스를 사용하여, DSA에 저장된 관광 정보를 DUA의 요청에 의한 관광 정보를 DSA에서 검색하여 정상적인 결과가 획득됨을 확인하였다.

6. 결 론

현재 개발, 운용되고 있는 정보 통신 서비스들은 각 서비스마다 서비스에 필요한 정보를 독립적으로 저장, 관리하고 있어 비효율적이며, 동일한 관리 기능의 개발에 많은 불필요한 노력을 기울이고 있는 실정이다. 이에 따라 서비스에 필요한 모든 정보를 통합적으로 관리할 수 있는 디렉토리 시스템의 필요성이 증대되고 있으며, 국내 관련 기관 및 전문가를 중심으로 X.500 표준을 바탕으로 한 디렉토리 S/W의 개발 및 디렉토리의 구축이 적극적으로 논의되고 있다. X.500 디렉토리 모델은 객체지향 개념을 기반으로 실세계에 존재하는 모든 객체를 그 대상으로 하여 모델링할 수 있도록 설계되었기 때문에 어떤 특정한 정보 통신망이나 서비스에 국한되지 않고, 서비스에 관련된 모든 정보를 저장, 관리할 수 있으므로 다양한 분야에 폭넓게 활용될 수 있는 OSI 프로토콜이다.

본 논문에서는 이러한 X.500 프로토콜을 이용하여 관광 정보를 위한 디렉토리 스키마를 설계하고, 그 결과를 ASN.1으로 정의하였으며, 이에 적합한 지역적 정보를 갖게 하는 DIT 구조를 제안하였다. 또한 X.500 디렉토리를 이용하여 설계한 디렉토리 시스템이 지역적인 구조를 갖는 정보 통신 응용 서비스가 적용될 수 있음을 보였다. 이 디렉토리 시스템은 조직적인 구조와 지역적 구조를 갖는 정보 서비스를 모두 포함하며, 개방형 시스템으로 설계하였기 때문에 정보 통신 응용 서비스로서 확장이 용이하다. 향후 연구는 본 논문에서 정의한 디렉토리 시스템을 구현과 초고속 통신망에서 이용될 수 있는 정보 통신 응용 서비스로서 개발이 필요하다.

참 고 문 헌

[1] 우종식, 안순신, "디렉토리 서비스," 정보과학회지, Vol.8, No.6, 1990.

[2] 이재광, 이용준, "X.500 디렉토리 정보보호," 한국통신정보보호학회지, Vol.4, No.3, 1994.

[3] 이용준, 박문성, 오주병, "X.500 디렉토리 표준," 주간기술동향, 한국전자통신연구소, 1994.

[4] 이용준, 박문성, 오주병, "관광 정보를 위한 스키마," 한국정보과학회, 제21회 추계발표회, 1994. 10.

[5] 박문성, 오주병, 김혜규, 진병운, 박성열, "AIN에서 SCF/SDF 인터페이스에 X.500 적용을 위한 AIN ADF Server 설계," 한국정보처리논문지 Vol.2, No.5, 1995. 9.

[6] 오주병, 박문성, 이용준, 진병운, 박성열, "The Application of X.500 DAP/DSP for the Distributed Processing in AIN," ICEIC'95, 1995. 8.

[7] CCITT F.500, "Directory Service: Operations and Definition of Service," CCITT, 1992.

[8] CCITT X.500, "The Directory: Overview of Concepts, Models, and Services," CCITT, 1992.

[9] CCITT X.501, "The DIrectory: The Models," CCITT, 1992.

[10] CCITT X.511, "The DIrectory: Abstract Service Definition," CCITT, 1992.

[11] CCITT X.519, "The Directory: Protocol Specification," CCITT, 1992.

[12] CCITT X.520, "The Directory: Selected Attribute Types," CCITT, 1992.

[13] CCITT X.521, "The Directory: Selected Object Class," CCITT, 1992.

[14] ISO 8824, "Information Processing System-Open System Interconnection-specification of Abstract Syntax Notation One (ASN.1)," ISO/IEC JTC1, 1987.

박 문 성

1993년 숭실대학교 대학원 전자공학과(석사)
1983년~현재 한국전자통신연구소, 정보공학연구실(연구원)
관심분야: 디렉토리 서비스, 객체지향 시스템, CSCW

오 주 병

1992년 충남대학교 공과대학 컴퓨터공학과(학사)
1994년 충남대학교 대학원 컴퓨터공학과(석사)
1994년~현재 한국전자통신연구소 정보공학연구실(연구원)
관심분야: 디렉토리 서비스, 객체지향시스템, 멀티미디어

양 해 철

1990년 국립대전산업대학교 전자계산학과 졸업(학사)
1996년 한남대학교 대학원 전자계산공학과 졸업(석사)
1989년~1993년 국제특수금속(주) 전산실 근무
1993년~현재 대전전문대학 전자계산소 근무
관심분야: 디렉토리 서비스, 컴퓨터통신 정보보호

이 용 준

1984년 광운대학교 전산과(학사)
1987년 연세대학교 대학원 전산과(석사)
1993년 정보처리기술사(전자계산조직응용)
1984년~현재 한국전자통신연구소 정보자동화연구실(선임연구원)
관심분야: 통신망 정보보호, 객체지향 개발방법론, 데이터베이스 설계

이 재 광

1984년 광운대학교 전산과(학사)
1986년 광운대학교 대학원 전산과(석사)
1993년 광운대학교 대학원 전산과(박사)
1986년~1993년 군산전문대학 전자계산학과 부교수
1993년~현재 한남대학교 전자계산공학과 조교수
1994년~현재 한국정보처리학회 학회지 편집위원
1995년~현재 한국통신정보보호학회 학회지 편집위원
관심분야: 컴퓨터네트워크, 컴퓨터통신 정보보호, 네트워크 관리