

# 인터넷 메타데이터 검색 및 관리 시스템의 설계 및 구현

## Design and Implementation of the Internet Metadata Retrieval and Management System

이원석(Won-Suk Lee)\*\*, 박경용(Kyoung-Yong Park)\*\*,  
양영종(Young-Jong Yang)\*\*\*, 정호택(Hyo-Taek Jeong)\*\*\*,  
김순용(Soon-Yong Kim)\*\*\*, 이규철(Kyu-Chul Lee)\*\*\*\*

### 목 차

- |                              |                              |
|------------------------------|------------------------------|
| 1. 서론                        | 4. 인터넷 메타데이터 검색 및 관리 시스템의 구현 |
| 2. 인터넷 메타데이터 DTD 정의          | 4.1 메타데이터의 로딩                |
| 3. 인터넷 메타데이터 검색 및 관리 시스템의 설계 | 4.2 사용자 인터페이스의 구현            |
| 3.1 전체 시스템의 구조 설계            | 4.3 메타데이터 검색 기능의 구현          |
| 3.2 각 모듈간의 인터페이스 설계          | 5. 결론                        |

### 초 록

인터넷은 디지털 형태의 다양한 정보를 신속하게 전달한다는 점에서 매우 각광을 받고 있다. 그러나, 대량의 인터넷 정보 중에서 자신이 원하는 정보 자원을 정확하게 많이 얻어 낼 수 있는 방법의 제공은 매우 어려운 문제이다.

본 논문에서는 인터넷 정보 자원의 메타데이터를 관리하기 위해 SeriCore라는 SGML 형식의 메타데이터 DTD(Document Type Definition)를 정의하고, 이들 메타데이터를 효율적으로 저장하고 검색, 관리할 수 있는 시스템을 설계, 구현하였다.

### ABSTRACT

The Internet has gained popularity because it provides various information in digital format. However, it is very difficult to provide methods which make it possible to search adequate information resource from bulk of Internet information. In this paper, we define a SGML metadata DTD, named SeriCore, and design and implement a metadata retrieval and management system which can store, retrieve and manage the Internet metadata efficiently.

\* 본 연구는 1996년 시스템공학연구소의 연구용역비에 의하여 수행되었음.  
\*\* 충남대학교 컴퓨터공학과 석사과정  
\*\*\* 시스템공학연구소 연구원  
\*\*\*\* 충남대학교 컴퓨터공학과 부교수  
접수일자 1997년 5월 20일

## 1. 서론

최근 들어, 컴퓨터 처리 속도의 향상, 통신 기술의 획기적인 발전 등으로 인터넷(internet)에 대한 사용이 폭발적으로 증가하고 있다. 현재 인터넷에서는 HTML(HyperText Markup Language)에 기반한 하이퍼미디어 문서 형태의 WWW(World-Wide Web) 정보, 소프트웨어 정보를 제공하는 Archie, 특정 기관의 정보를 체계적으로 정리해 놓은 Gopher, 공용으로 활용 가능한 FTP 정보, WAIS등 다양한 형태로 각종 분야에 대한 정보를 제공하고 있다.

이와 같은 인터넷 상에서 제공되는 정보는 기존에 인쇄되어진 도서나 문서 등과는 여러 가지 측면에서 다른 점을 가지고 있다. 첫째는 정보 제공자(information provider)가 인터넷과 접할 수 있는 모든 사람이 될 수 있으므로 인터넷 상의 정보들이 매우 다양하면서 광범위해졌다는 점이다. 둘째는 인터넷 상의 정보들이 텍스트나 이미지는 물론 그래픽, 오디오, 비디오 등을 포함하는 멀티미디어(multimedia) 정보의 형태로 전달되고 있다는 점이다. 셋째는 인터넷상의 모든 정보는 디지털(digital)화 되어진 것이므로, 필요한 정보들은 자신이 원하는대로 적절하게 가공하여 활용할 수 있다는 점이다.

위와 같은 인터넷 상의 정보들이 갖는 장점 때문에, 요사이 사용자들의 가장 큰 관심은 광범위하고 다양한 인터넷 정보

중에 자신이 원하는 정보를 얼마나 정확하게, 많이 얻어낼 수 있는가에 쏠려 있다.

이런 문제를 해결하기 위한 방법으로 대표적으로 쓰이고 있는 기술이 바로 인터넷 상의 각종 탐색 엔진(search engine)들이다. 대표적인 탐색 엔진들은 외국의 경우 Infoseek, Web Crawler, Lycos, Alta Vista, Exite 등이 있으며, 국내에서도 심마니와 같은 수십개의 탐색 엔진들이 제공되고 있다(정영미, 김성은, 1997).

각 탐색 엔진들은 로봇 에이전트(robot agent)를 이용하여, 인터넷 상에 산재한 정보들을 수집하고, 이들 정보의 내용을 자동으로 색인(indexing)하여, 사용자가 찾고자 하는 키워드를 주면 이를 구축한 색인을 이용해 탐색한 뒤 필요한 정보의 위치 정보와 그 정보의 간단한 요약 내용들을 제공해 준다. 이와 같은 탐색 엔진들이 제공해 주는 내용에 대한 색인 정보, 위치 정보, 요약 정보들은 인터넷 데이터들에 대한 데이터, 즉 메타데이터(metadata)가 된다.

이 메타데이터는 보통 화일 시스템에서 제공되는 디렉토리(directory) 정보, 데이터베이스 시스템의 카탈로그(catalog) 정보, CASE 도구에서 이용되는 데이터사전(data dictionary) 등과 유사하게 정보 자원(information resource)을 제대로 활용하기 위해서는 필수 불가결한 요소이다.

그러나 현재 인터넷 상에서 원하는 정보 자원을 찾기 위해 이용되는 각종 탐색 엔진은 정확한 메타데이터를 제공해 주지

못하고, 사용자가 필요한 메타데이터를 부가하기 힘들다는 단점 때문에 궁극적인 메타데이터 검색 및 관리를 위한 시스템으로 활용하기 힘들다(Lagoze, 1996; Weibel, 1995).

본 논문에서는 인터넷 정보 자원의 메타데이터를 검색 및 관리하기 위해 메타데이터를 SeriCore라는 SGML(Standard Generalized Markup Language) DTD(Document Type Definition)로 정의하고, 인터넷 정보자원에 대한 메타데이터를 효율적으로 저장하고 검색, 관리할 수 있는 시스템을 설계 및 구현하였다. 본 논문의 구성은 다음과 같다. 2장에서는 인터넷 메타데이터 DTD 정의에 대하여 기술한다. 3장에서는 메타데이터 검색 및 관리 시스템의 설계 내용을 기술하고, 4장에서는 구현내용을 기술한다. 마지막으로 5장에서는 본 연구의 결론에 대해서 기술한다.

## 2. 인터넷 메타데이터 DTD 정의

메타데이터를 표준화된 형태로 기술하기 위해서는 메타데이터를 정의하는 골격(framework)이 필요하다.

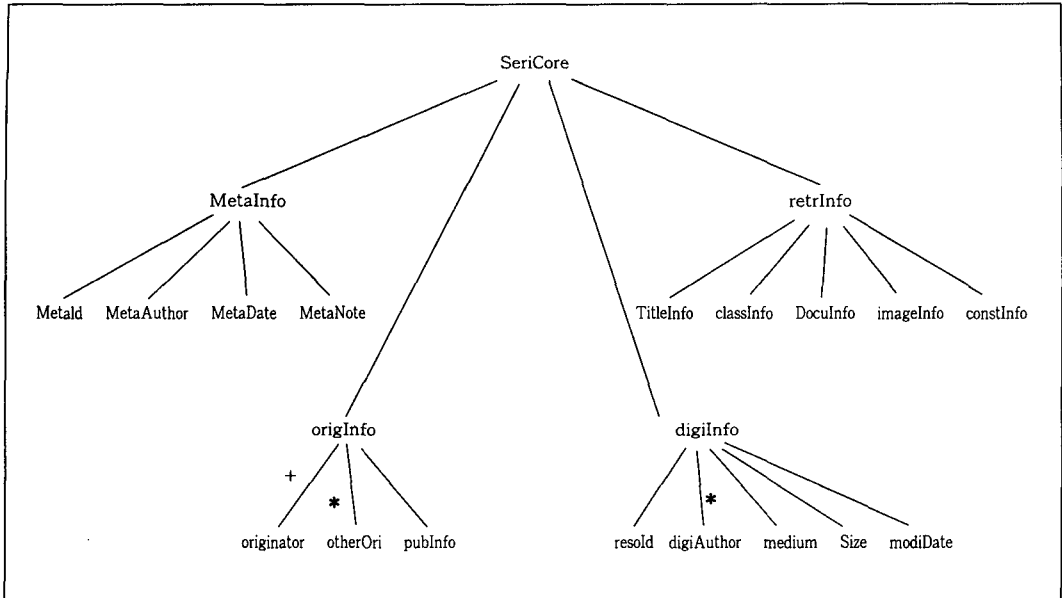
이를 위해 1995년 3월에 처음으로 미국 Ohio Dublin에서 인터넷 정보 자원의 메타데이터에 대한 워크샵이 개최되었고, 이 회의에서 인터넷의 정보 자원에 대한 메타데이터가 최소한 가져야 할 요소(element)들을 SGML의 DTD로 정의하

였는데, 이를 Dublin Metadata Core라 명칭하였다(Weibel, 1995).

그 후, 1996년 4월에 메타데이터에 대한 두번째 회의가 Warwick 대학에서 개최되었는데, 이 회의에서는 첫번째 회의 결과인 Dublin Core를 포함하면서 기타 다른 형식의 메타데이터의 기술까지도 포함하도록 DTD를 확장하여 Warwick DTD를 정의하였다(Hakala, 1996).

본 연구에서는 인터넷 메타데이터 기술을 위해 Warwick DTD를 기반으로 하고, 시스템공학연구소(System Engineering Research Institute:SERI)에서 인터넷 정보자원을 관리하기 위해 추가로 필요한 요소들을 포함시켜 확장 정의하였다. 이것을 SeriCore DTD 라고 명칭하였다. 정의된 SeriCore DTD의 내용은 <부록>에 수록하였다. SeriCore DTD에서 정의한 메타데이터는 크게 메타정보(metaInfo), 원시정보(origInfo), 디지털정보(digiInfo), 검색정보(retrInfo)의 4가지로 나누어 정의되어있다. 메타정보는 메타정보의 메타정보를, 원시정보는 원저자 정보나 발표된 출처에 대한 정보를, 디지털정보는 망으로 접근할 수 있는 자원들에 대한 식별정보, 매체정보, 작성자정보를, 검색정보는 제목정보, 분류정보, 문서구조 내용정보, 이미지 내용정보, 제한정보를 의미한다.

<그림 1>은 SeriCore DTD 구조를 레벨 3까지만 표현한 것이다. '+' 와 '\*' 기호는 SGML의 Occurrence Indicator를 표현한 것으로, '+'는 요소가 하나 혹은 그 이상이 올 수 있음을 나타내고, '\*'은 나



〈그림 1〉 SeriCore 메타데이터 DTD의 구조

오지 않거나 하나 이상 나올 수 있음을 의미한다.

〈그림 1〉의 각 메타데이터 요소들에 대한 내용은 아래와 같다.

■ 메타정보(metaInfo)에 포함된 요소

- metaId : 대상 메타 정보의 식별자.
- metaAuthor : 대상 메타 정보의 작성자.
- metaDate : 대상 메타정보가 만들어진 날짜정보.
- metaNote : 대상메타정보에 대해 메타정보 작성자가 제공하는 부수적인 정보.

■ 원시정보(origInfo)에 포함된 요소

- originator : 대상 정보 자원의 원저자.
- otherOri : 대상 정보 자원에 중요한 기여를 한 여타 다른 사람.
- pubInfo : 대상 원시 정보가 게재된 원전 명, 원전을 게재한 출판사, 원전이 출판된 날짜, 원전이 게재된 행사나 학회명, 원전의 페이지 수 등에 대한 정보.

■ 디지털정보(digiInfo)에 포함된 요소

- resoId : 디지털 정보에 대한 식별자.
- digiAuthor : 대상 정보 자원의 매체 작성자.
- medium : 대상 정보 자원의 정보

- 형태(문서, 이미지 등).
- Size : 대상 정보 자원의 크기.
- modiDate : 대상 디지털 정보가 만들어진 날짜에 대한 정보.

■ 검색정보(retrInfo)에 포함된 요소

- titleInfo : 대상 정보 자원의 제목, 부제목, 제목을 기술한 언어 등에 대한 정보.
- classInfo : 대상 정보 자원의 문서상의 분류, 그림상의 분류 등에 대한 정보.
- docuInfo : 대상 정보 자원의 요약문, 색인어, 목차, 의미있는 그림 및 테이블, 참조문헌 등에 대한 정보.
- imageInfo : 대상 정보 자원의 이미지에 대한 서술, 캡션, 공간 지리 등에 대한 정보.

- constInfo : 대상 정보 자원에 대한 권리, 접근 제한, 사용 제한에 대한 정보.

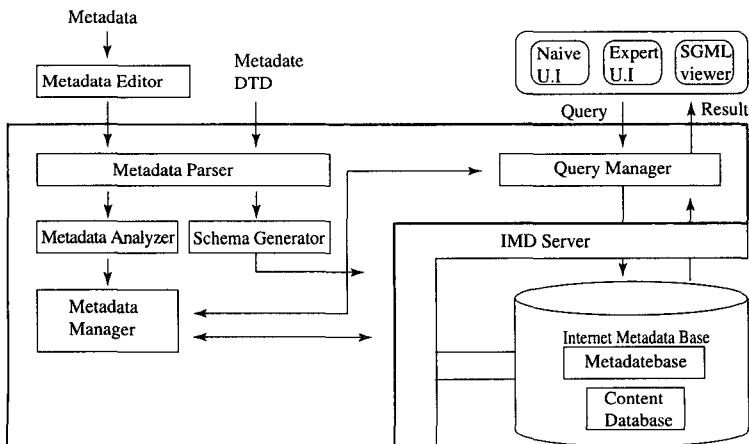
인터넷 메타데이터 인스턴스(instance)들은 SeriCore DTD에서 정의한 규칙에 맞는지 분석, 검증된 후 데이터베이스에 저장되고, 사용자들은 이를 검색, 사용할 수 있다.

### 3. 인터넷 메타데이터 검색 및 관리 시스템의 설계

#### 3.1 전체 시스템의 구조 설계

본 연구에서는 디지털 정보의 저작/생성/제공 시, 인터넷 상에서 제공되는 각종 디지털 데이터를 정보 부품으로서 재활용하기 위해, 인터넷 데이터에 대한 메타데

<그림 2> 인터넷 메타데이터 검색 및 관리 시스템의 구성도



이타를 효율적으로 저장하고 검색, 관리할 수 있는 시스템을 개발하였다.

본 연구에서 개발한 인터넷 메타데이터 검색 및 관리 시스템(IMDREMS: Internet MetaData REtrieval and Management System)의 기본 구성도는 <그림 2>와 같다.

IMDREMS는 인터넷 상에서 채취한 원래의 디지털 데이터 자체 뿐만 아니라 차후 재활용을 위해 관련 인터넷 데이터를 쉽게 찾을 수 있도록 하는 다양한 단서 정보, 즉 메타데이터를 관리할 수 있으며, 이를 기반으로 다양한 형태의 질의를 제공할 수 있다.

<그림 2>에서 제시한 IMDREMS 시스템은 다음과 같은 8가지 모듈(module)로 구성되며, 각 구성 모듈의 기능은 다음과 같다.

■ 메타데이터 편집기(Metadata Editor)

메타데이터 편집기에서는 주어진 메타데이터의 DTD에 따라 메타데이터 요소들의 값을 입력할 수 있는 기능을 제공한다. 이때 각 요소들을 구분하기 위한 태그(tag)들을 자동적으로 템플릿(templet) 형태로 제공함으로써 메타데이터를 생성하는 사용자의 오류를 방지하고, 편의성을 제공한다.

■ 메타데이터 파서(Metadata Parser)

메타데이터 파서의 역할은 크게 두가지

로 나누어 볼 수 있는데, 첫째는 정의한 메타데이터 DTD의 오류를 검사하는 기능과, 둘째는 편집기를 통해 입력된 메타데이터 인스턴스의 오류를 검사하는 기능이다.

■ 스키마 생성기(Schema Generator)

분석된 메타데이터를 데이터베이스에 체계적으로 저장 관리하기 위해서는 효율적인 스키마의 구축이 필요하다. 스키마 생성기는 파싱된 DTD 정보를 기반으로 하부 객체 지향 데이터베이스에 저장 가능한 형태로 필요한 클래스(class) 및 메소드(method)를 자동적으로 생성한다.

■ 메타데이터 분석기(Metadata Analyzer)

메타데이터 파서는 단지 메타데이터의 오류만을 검사해 주기 때문에, 입력된 메타데이터를 각 요소별로 저장하기 위해서는 각 요소들을 분석하여 분리시키는 기능이 필수적이다. 이와 같은 기능을 수행하는 메타데이터 분석기는 메타데이터 파서의 결과를 입력으로 받아, 각 요소를 분리한 뒤 메타데이터 관리기에게 각 요소에 해당하는 객체들을 생성하도록 전달하게 된다.

■ 메타데이터 관리자(Metadata Manager)

메타데이터 분석기에 의해 분해된 각 메타데이터 요소들은 생성된 스키마에 따라 저장될 수 있도록 객체(object) 형태로

만들어져야 한다. 메타데이터 관리기는 이와 같은 객체의 생성 뿐만 아니라, 메타데이터 요소 객체 상호간의 구조적 관계성을 파악 유지하여, 객체의 변경이나 삭제에 대한 관리까지도 담당한다. 물론 인터넷에서 채취된 원래 인터넷 데이터를 객체화하여 저장하고 관리하는 기능도 메타데이터 관리기가 수행하는 역할 중에 중요한 부분이다.

#### ■ 질의 관리기(Query Manager)

사용자로부터 질의를 받아, 이를 하부의 인터넷 메타데이터 서버에서 사용되는 질의언어로 변환하여 전달하고, 메타데이터 베이스를 검색한 결과를 사용자에게 이해하기 쉬운 형태로 변환하여 주는 역할을 담당한다.

#### ■ 인터넷 메타데이터 서버(Internet Metadata Server)

메타데이터 서버는 위에서 설명한 인터넷 메타데이터 관리를 위해 각 모듈을 구현하기 위한 기반이 되는 시스템이다. 메타데이터가 가진 복잡한 구조를 제대로 표현, 처리하기 위해서 객체 지향 데이터 모델을 지원하는 데이터베이스 서버인 UniSQL/X(UniSQL, 1996)를 기반으로 구현하였다.

#### ■ 인터넷 메타데이터베이스(Internet Metadata Base)

인터넷 메타데이터베이스의 구성은 크게 원래의 인터넷 데이터의 내용을 담고

있는 인터넷 내용 데이터베이스(Content Database), 메타데이터로 구성되는 메타데이터베이스(Metadatabase)의 두 가지 부분으로 구성된다.

### 3. 2 각 모듈간의 인터페이스 설계

본 절에서는 3.1절에서 설계한 인터넷 메타데이터 검색 및 관리 시스템의 각 모듈간에 서로 어떻게 상호 작용하는지 그 인터페이스 설계 내용을 기술한다.

#### 3. 2. 1 스키마의 생성

메타데이터의 DTD를 입력으로 받아 SGML 파서로 오류가 없는 지를 검사하고, 오류가 없으면 DTD를 분석하여 스키마를 정의할 때 필요한 정보들을 추출한다. 이 정보를 기반으로 스키마를 정의하고 생성한다.

#### 3. 2. 2 객체 생성

객체를 생성하는데 있어서 메타데이터 분석기는 SGML 파서의 결과를 입력으로 받아 메타데이터 인스턴스를 분석하여, 각 논리적 요소를 분리한다. 이 분석된 결과는 메타데이터 관리자에게 전달되어 객체를 생성하는데 사용된다.

메타데이터 관리자가 객체를 생성하는 과정은 먼저 메타데이터 분석기에 의해 분리된 각 메타데이터 요소를 객체(object)로 생성한다. 각각의 객체는 객체 식별자(OID)를 가지게 되는데, 이 객체 식별자는 상위 요소(super element)를 구성

하기 위해 사용된다.

### 3. 2. 3 검색

인터넷 메타데이터베이스에 저장되어 있는 메타데이터를 검색하기 위한 사용자 인터페이스로서 웹 브라우저(Web Browser)를 사용한다. 질의 관리자는 사용자가 입력한 검색어를 분석하여 메타데이터 서버에 적절한 질의문을 생성하고, 생성된 질의문을 메타데이터 서버에 전달한다. 메타데이터 서버가 질의 결과를 반환하면 HTML 형태로 변환하여 웹 브라우저로 보여준다.

질의문은 메타데이터 서버의 SQL문을 따른다.

### 3. 2. 4 브라우징 및 항해

브라우징(browsing)은 검색의 다음 단계로서 사용자가 검색한 메타데이터의 내용을 찾아보는 과정이다. 브라우징 결과에서 하이퍼링크(hyper link)가 있으면, 그곳을 통하여 다른 메타데이터로 이동할 수 있는데, 이 과정이 항해(navigation)이다. 현재는 메타데이터의 브라우징 및 항해를 위해서 SoftQuad사의 Panorama Pro(SoftQuad, 1997)라는 SGML Browser를 사용하고 있다.

## 4. 인터넷 메타데이터 검색 및 관리 시스템의 구현

### 4. 1 메타데이터의 로딩

시스템에 로딩(loading)되는 메타데이터 인스턴스는 메타데이터 파서를 이용해 DTD와 일치되는지 메타데이터 자체의 오류를 검사하게 된다. 만일 오류가 있으면 오류 메시지와 함께 메타데이터의 로딩을 종료하고, 오류가 없으면 메타데이터 분석기가 사용할 수 있도록 파싱(parsing)된 정보를 파일로 저장한다. 메타데이터 분석기는 이 파일을 읽어들이어 각 요소를 분리하고 메타데이터 관리기가 각 요소에 대한 객체들을 생성할 수 있도록 정보를 넘겨준다.

〈그림 3〉은 파싱된 메타데이터를 읽어들이어 데이터베이스에 저장하는 주요 원시 코드이다. 즉, Meta Information, Original Resource Information, Digital Resource Information, Retrieval Information 각각의 요소에 대하여 시작 위치에서 해당하는 함수를 호출하고 각 객체에 대한 객체식별자를 리턴 받는다. metaInfo\_obj, origInfo\_obj, digiInfo\_obj, retrInfo\_obj, instance\_obj는 모두 객체식별자이며, 이들을 이용하여 메타데이터객체를 생성한다.

### 4. 2 사용자 인터페이스의 구현

본 시스템에서는 SeriCore DTD 구조를 파악하고 있는 사용자를 숙련자(expert)로 정의하고, 그렇지 못한 사용자를 초보자(naive)라고 정의하여 두 가지 종류의 사



〈그림 3〉 SeriCore 객체의 생성

```

while (file_get_string()) {
if (!strcmp(buffer, "(METAINFO)") {
    metaInfo_obj = create_metaInfo();
}
else if (!strcmp(buffer, "(ORIGINFO)") {
    origInfo_obj = create_origInfo();
}
else if (!strcmp(buffer, "(DIGIINFO)") {
    digiInfo_obj = create_digiInfo();
}
else if (!strcmp(buffer, "(RETRINFO)") {
    retrInfo_obj = create_retrInfo();
}
else if (!strcmp(buffer, ")SERICORE")) {
    instance_obj = create_instance(file_name);

    EXEC SQLX INSERT INTO SeriCore(metaInfo, origInfo, digiInfo, retrInfo, instance) values(:metaInfo_obj,
:origInfo_obj, :digiInfo_obj, :retrInfo_obj, :instance_obj);
}
}
    
```

용자 인터페이스를 제공한다.

4. 2. 1 초보자를 위한 사용자 인터페이스

SeriCore DTD에 대해 아무런 지식 없이도 자신 원하는 메타데이터를 찾을 수 있도록 SeriCore DTD의 요소중 일반적으로 검색 할 때 가장 많이 사용하는

〈그림 4〉 초보자용 사용자 인터페이스

제목	:	<input type="text"/>
원저자	:	<input type="text"/>
키워드	:	<input type="text"/>
출판원전	:	<input type="text"/>
출판사	:	<input type="text"/>
관련행사	:	<input type="text"/>
매체작성자	:	<input type="text"/>
자료분류	:	TOTAL ▼
조합연산자	:	AND ▼
		<input type="button" value="검색"/> <input type="button" value="다시입력"/>

---

SeriCore  
System Engineering Research Institute

〈그림 5〉 숙련자용 사용자 인터페이스

질의문

---

▼ Search Reset

---

<i>metaInfo</i>	+	메타정보
<i>origInfo</i>	+	원시정보
<i>DigiInfo</i>	+	디지털정보
<i>retrInfo</i>	+	검색정보

---

SeriCore  
System Engineering Research Institute

제목, 원저자, 키워드, 출판원전, 출판사, 관련행사, 매체작성자, 자료분류에 대해서만 검색을 할 수 있도록 구성하였다. 〈그림 4〉는 HTML의 <FORM> 태그를 이용하여 만든 초보자를 위한 사용자 인터페이스이다.

#### 4. 2. 2 숙련자를 위한 사용자 인터페이스

숙련자는 SeriCore DTD의 구조와 각 요소들이 의미하는 것을 알고 있으므로 원하는 정보를 찾는데 좀 더 높은 수준의 질의를 할 수 있어야 한다. 〈그림 5〉는 이러한 요구를 만족시키는 숙련자를 위한 사용자 인터페이스이다. 사용자가 SeriCore DTD의 모든 요소에 질의할 수 있도록 DTD 구조가 사용자의 선택에 따라 동적으로 확장, 축소된다. 사용자가 원하는 요소를 선택하면 FILL-OUT-FORM이 나타나고, 검색어를 주면 상단

에 있는 바(bar)에 질의문이 자동으로 생성된다. 고정된 값만을 가져야 하는 요소에 대해서는 FILL-OUT-FORM 대신 선택 버튼으로 보여지고 사용자는 선택만 하도록 하였다. 요소 옆에 '+' 기호가 있는 것은 서브 요소(sub element)가 있음을 나타낸다.

#### 4. 3 메타데이터 검색 기능의 구현

본 시스템은 초보자용과 숙련자용의 두 가지의 사용자 인터페이스에 대해 각각의 질의 관리기를 구현하였다. 질의 관리기의 역할은 사용자의 검색어를 입력받아 본 시스템의 메타데이터 서버에 적합한 질의문을 만들고, 수행한 결과를 웹 브라우저로 보여주는 시스템의 가장 핵심이 되는 부분이다.

초보자용 질의 관리기에서는 먼저 각각의 요소에 대한 검색어가 연산자('&' 나

'')와 함께 여러 검색어로 되어 있는지를 검사하고, 있으면 연산자와 검색어를 분리해낸다. 이렇게 분석된 정보와 8개 요소에 대한 연산자 정보를 이용해서 SQL문의 from절과 where절을 각각 나누어 만든다. 이렇게 나누어 만드는 이유는 set 타입의 애트리뷰트에 대해 질의를 할 경우 메타데이터 서버의 set derived table을 이용해야 하고, 이것은 from절을 변화시키기 때문이다. 이렇게 만들어진 from절과 where절을 이용해서 완전한 질의문으로 만들고 메타데이터 서버로 전달하면, 질의문의 수행된 결과를 반환하게 되는데, 이것을 사용자의 웹 브라우저로 보여주게 된다.

숙련자용 질의 관리기능은 두 부분로 나누어져 있는데, 한쪽 부분은 사용자 인터페이스쪽에 포함되어져 있고 나머지 부분은 질의 관리자에 구현되어 있다. 사용자 인터페이스쪽에서는 사용자의 검색어를 받아서 초보자용 질의 관리기에서와 같이 분석을 하고 from절과 where절을 각각 만든다. 각 요소에 대한 조건절 간의 구분은 ';'로 표현하게 된다. 질의 관리자는 이렇게 만들어진 from절과 where절을 받아서 ';'부분을 적절한 연산자(AND 또는 OR)로 바꾸어주고 완전한 질의문을 만든다. 질의문을 메타데이터 서버에 전달하고 결과를 보여주는 과정은 초보자용과 동일하다.

## 5. 결론

광범위하고 다양한 인터넷 정보 중에서 자신이 원하는 정보 자원을 정확하게 많이 찾아 낼 수 있는 방법의 제공은 사용자가 인터넷 정보 자원을 재활용하여 사용할 수 있다는 점에서 매우 중요한 문제이다.

따라서, 인터넷 정보 자원을 효율적으로 이용하기 위해서는 정보 자원에 대한 내용 및 위치 정보 등의 메타데이터의 관리 및 정확한 해석을 제공해 주는 것이 필요하다.

현재 인터넷 상에서 원하는 정보 자원을 찾기 위해 이용되는 각종 탐색 엔진은 정확한 메타데이터를 제공해 주지 못하고, 멀티미디어 데이터와 사용자가 필요한 메타데이터를 부가하기 힘들다는 단점에 공극적인 메타데이터 검색 및 관리를 위한 시스템으로 활용하기 힘들다.

더우기 디지털 정보의 저작/생성/제공시, 인터넷 상에서 제공되는 각종 디지털 데이터를 정보 부품으로서 재활용하기 위해서는 인터넷 데이터에 대한 메타데이터를 효율적으로 저장하고 검색, 관리할 수 있는 시스템이 필요하다.

본 연구에서는 이와 같은 기능을 제공하는 인터넷 메타데이터 검색 및 관리 시스템을 개발하였다.

개발된 시스템은 현재 도서관 시스템에서 주 대상으로 하는 서지 정보 자원 관리의 범주를 뛰어 넘어, 인쇄매체는 물론 인터넷 자원까지를 관리할 수 있는 메타데이터 관리의 새로운 참고 시스템이 될 수 있으며, 디지털 도서관 구축에 필요한

기반 기술로서 활용될 수 있을 것으로 생각된다.

## 참고문헌

- 정영미, 김성은. 1997. "WWW 탐색도구의 색인 및 탐색 기능 평가에 관한 연구", 한국문헌정보학회지 31(1): 153-184
- Hakala, J., Husby, O. and Koch, T. 1996. "Warwick Framework and Dublin Core Set Provide A Comprehensive Infrastructure for Network Resource Description", (URL: <http://www.bibsys.no/warwick.html>)
- Knight, J. & Hamilton, M. 1996. "A MIME implementation for the Warwick Framework", (URL: <http://weeble.lut.ac.uk/MIME-WF.html>)
- Lagoze, C. 1996. "The Warwick Framework: A Container Architecture for Diverse Sets of Metadata", D-Lib Magazine July/August
- Lasher, R. & Cohen, D. 1995. RFC 1807: "A Format for Bibliographic Records", (URL: <http://ds.internic.net/rfc/rfc1807.txt>)
- Miller, P. 1996. "Archaeology Data Service", (URL: <http://www.ncl.ac.uk/~napml/ads/metadata>)
- SoftQuad, Inc. 1997. Panorama Pro User Manual
- UniSQL, Inc. 1996. UniSQL/X User Manual
- Weibel, S. 1995. "Metadata: The Foundations of Resource Description." D-Lib Magazine July (URL: <http://www.dlib.org/dlib/July95/07weibel.html>)
- Weibel, S. 1996. "A Proposed Convention for Embedding Metadata in HTML" A working group report from the W3C Distributed Indexing Workshop (URL: <http://www.oclc.org:5046/~weibel/html-meta.html>)

## 〈부록〉 SeriCore DTD의 정의

```

<!-- ***** -->
<!-- SeriCore DTD for Metadata Description of Technical Documents -->
<!-- Edited by yjyang/Open S/W Lab/SERI, '97. 1. 30. -->
<!-- ***** -->

<!-- ***** Definition Part ***** -->
    <!-- ***** Entity Naming Convention ***** -->

<!-- Prefix (where used)
    m. = content model or declared content
    a. = attribute definition
        Suffix
    .Scheme = description method
    .d = elements whose content has same model as defaults
-->

    <!-- ***** Scheme Definition ***** -->
<!-- ANSIX3.30 ::= yyyymmdd ANSIX3.43 ::= hhmss.f -->
<!ENTITY % Date.Scheme
    "ANSIX3.30 | ANSIX3.43 | ANSIX3.51 | Others" >
<!ENTITY % Id.Scheme
    "URN | URL | LCCN | ISBN | ISSN | SICI | MessageID | FPI | Others" >

    <!-- ***** Special Elements ***** -->
<!ENTITY % id.d "metaId | resoId | figureId | biblioId" >
<!ENTITY % date.d "metaDate | pubDate | modiDate | timePeri" >

    <!-- ***** Models ***** -->
<!ENTITY % m.date "(formDate | textDate)" -- date model -->

    <!-- ***** Common Element ***** -->
<!-- ENTITY % plainText "namePub | publisher | volNumEd | pages" -->
    <!-- ***** Attribute Models ***** -->
<!ENTITY % a.language

```

```

        "language (Korean | English | Japanese | Others) #IMPLIED" >
<!ENTITY % a.docCategory
        "Technical | Research | Paper | Manual | OtherDoc" >
<!ENTITY % a.imgCategory
        "Graphic | Photo | Drawing | OtherImg" >
<!ENTITY % a.mediType
        "Document | Image | OtherType" >
<!ENTITY % a.mediForm
        "html | ps | doc | ppt | txt | hwp | bmp | pcx | gif | jpg |OtherForm" >
<!ENTITY % a.metaAuthor
        "type (Originator| DigiAuthor| WebBuilder| IP| Others) #IMPLIED" >
<!ENTITY % a.originator
        "type (MainAuthor| CoAuthor| Others) #IMPLIED" >
<!ENTITY % a.otherOri
        "type (Editor|Translator|Illustrator|Photographer|Others) #IMPLIED" >
<!ENTITY % a.digiAuthor
        "type (Originator|MetaAuthor|WebBuilder|IP|Others) #IMPLIED" >

        <!-- ***** Common Element Content Model ***** -->
<!ELEMENT (%id.d;) - - (#PCDATA) >
<!ATTLIST (%id.d;)
scheme (%Id.Scheme;) #IMPLIED
systemId CDATA #IMPLIED >

        <!-- ***** Date Group ***** -->
<!ELEMENT (%date.d;) - - %m.date; >

        <!-- ***** Element date ***** -->
<!ELEMENT formDate - - CDATA >
<!ATTLIST formDate scheme (%Date.Scheme;) #IMPLIED >
<!ELEMENT textDate - O (#PCDATA) >

        <!-- ***** Plain Text ***** -->
<!-- ELEMENT (%plainText;) - - (#PCDATA) -->
<!-- ***** End of Definition Part ***** -->

```

```

<!-- ***** Main Part ***** -->
<!ELEMENT SeriCore - - (metaInfo, origInfo, digiInfo, retrInfo) >

<!-- *** Part 1: metaInfo = Meta Information -->
<!ELEMENT metaInfo - - (metaId, metaAuthor, metaDate?, metaNote?) >
<!ELEMENT metaAuthor - - (anAuthor) >
<!ATTLIST metaAuthor %a.metaAuthor; >
<!ELEMENT metaNote - - (#PCDATA)>

<!-- *** Part 2: origInfo = Original Resource Information -->
<!ELEMENT origInfo - - (originator+, otherOri*, pubInfo?) >
<!ELEMENT originator - - (anAuthor) >
<!ATTLIST originator %a.originator; >
<!ELEMENT otherOri - - (anAuthor) >
<!ATTLIST otherOri %a.otherOri; >

<!-- Publication Information -->
<!ELEMENT pubInfo - - (namePub, publisher?, pubDate?, nameEvent?, volNumbed?, pages?)>
<!ATTLIST namePub isbnCode CDATA #IMPLIED >
<!ELEMENT (namePub | publisher | nameEvent | volNumbed | pages) - - (#PCDATA) >

<!-- *** Part 3: digiInfo = Digital Resource Information -->
<!ELEMENT digiInfo - - (resoId,medium,size?,digiAuthor*,modiDate?) >
<!ELEMENT medium - - EMPTY >
<!ATTLIST medium
    meditype (%a.meditype;) #IMPLIED
    mediform (%a.meditype;) #IMPLIED >
<!ELEMENT size - - EMPTY >
<!ATTLIST size
    Kbytes NMTOKEN #IMPLIED
    pagesA4 NMTOKEN #IMPLIED >
<!ELEMENT digiAuthor - - (anAuthor) >
<!ATTLIST digiAuthor %a.digiAuthor; >

```

```

<!-- *** Part 4: retrInfo = Retrieval Information -->
<!ELEMENT retrInfo - -
(titleInfo, classInfo, (docuInfo | imageInfo), constInfo?) >

<!-- 4.1 : Title Information -->
<!ELEMENT titleInfo - - (title)+ >
<!ELEMENT title - - (mainTitle, subTitle?) >
<!ATTLIST title %a.language; >
<!ELEMENT (mainTitle | subTitle) - - (#PCDATA) >

<!-- 4.2 : Classification Information -->
<!ELEMENT classInfo - - (category, subjectTerm) >
<!ELEMENT category - - EMPTY >
<!ATTLIST category
    docCategory (%a.docCategory;) #IMPLIED
    imgCategory (%a.imgCategory;) #IMPLIED >
<!ELEMENT subjectTerm - - (subUnconTerm+ | subConTerm+) >
<!ELEMENT (subUnconTerm | subConTerm) - - (#PCDATA) >

<!-- 4.3 : Document Content Information -->
<!ELEMENT docuInfo - -
(abstract+, keyWord+, langMainText?, content*, figure*, biblio*, fundOrg?) >
<!ELEMENT abstract - - (#PCDATA) >
<!ATTLIST abstract %a.language; >
<!ELEMENT figure - - (figureDesc, figureId?) >
<!ELEMENT biblio - - (biblioDesc, biblioId?) >
<!ELEMENT (keyWord | langMainText | content | figureDesc | biblioDesc | fundOrg) -
- (#PCDATA) >

<!-- 4.4 : Image Content Information -->
<!ELEMENT imageInfo - - (resoDesc, addAttrVal*, caption*, spatial?, timePeri?) >
<!ELEMENT addAttrVal - - (addAttr, addVal) >
<!ELEMENT spatial - - (place*, coordinate?) >
<!ELEMENT coordinate - - EMPTY >
<!ATTLIST coordinate

```



```

WBC NMTOKEN #IMPLIED
EBC NMTOKEN #IMPLIED
NBC NMTOKEN #IMPLIED
SBC NMTOKEN #IMPLIED >
<!ELEMENT (resoDesc | addAttr | addVal | caption | place) - - (#PCDATA) >

<!-- 4.5 : Constraints Information -->
<!ELEMENT constInfo - - (copyright?, accessConst?, useConst?) >
<!ELEMENT (copyright | accessConst | useConst) - - (#PCDATA) >

<!-- ***** Common Part ***** -->
<!-- Common Compound Element : An Author -->
<!ELEMENT anAuthor - - (name, contactPnt?) >
<!ELEMENT name - O (#PCDATA) >

<!-- Common Compound Element : Contact Point -->
<!ELEMENT contactPnt - -
(position?, department?, orgName, addrKorean?, addrForeign?, postCode?,
country?, networkAdd?, hourServ?, telephone?, fax?) >
<!ELEMENT networkAdd - - (email?, homePage?) >
<!ELEMENT (position | department | orgName) - O (#PCDATA) >
<!ELEMENT (addrKorean | addrForeign | postCode | country ) - O (#PCDATA) >
<!ELEMENT (email | homePage | hourServ | telephone | fax) - O (#PCDATA) >

<!-- ***** End of SeiCore DTD ***** -->

```