

□특집□

차세대 웹 상에서의 문서 교환 및 검색을 위한 프레임워크

나 흥 석[†] 채 진 석^{††} 김 창 화^{†††} 백 두 권^{††††}

◆ 목 차 ◆

- | | |
|------------------------|------------------------------|
| 1. 서 론 | 4. 데이터 레지스트리 중심의 웹문서 교환 및 검색 |
| 2. 웹문서의 의미표현 | 5. 결 론 |
| 3. 웹문서 설명을 위한 메타데이터 구조 | |

1. 서 론

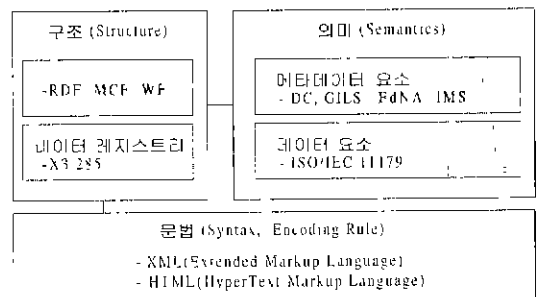
현재의 웹 환경에서 HTML은 문서 교환의 주된 표현 수단으로 이용된다. 그러나 한정된 태그 집합을 갖는 HTML은 문서를 비롯한 다양한 네트워크 자원을 효율적으로 교환 및 검색하기에는 한계가 있다. 또한, 제한된 표현력과 의미론의 부재로 인하여, 자원의 검색과 교환 시에 사람의 도움을 많이 필요로 한다.

이에 대한 하나의 해결방안으로 제시된 것이 XML(eXtensible Markup Language)이다. W3C(World Wide Web Consortium)에 의해서 개발된 XML은 ISO의 SGML을 인터넷에서 사용할 수 있도록 단순화한 버전이다[1]. XML을 이용하면 응용 프로그램 설계자는 웹문서에 대한 태그 집합과 구조를 생성할 수 있으며, 이들을 이용하여 문서, 데이터베이스, 객체, 카탈로그 또는 일반 응용 프로그램에서 데이터 교환에 사용되는 정보를 정의하고 기술할 수 있다[2].

그러나, XML은 문서의 구조를 표현하고 이해하는 데에서는 HTML의 한계를 극복하는 차세대 웹문서 표준으로 충분하지만, 문서내에 포함되어 있는 데이터의 의미에 대한 표현은 지원하지 않기 때문에, 문서 공유를 위한 교환 형식으로서는 부족하다. XML은 DTD(Document Type Definition)를 이용하여 단지 데이터의 문법적(구조적)인 정보만을 보여줄 뿐이며, 그 데이터에 대한 의미적인 정보를 표현할 수 없다.

따라서, XML로 대표되는 차세대 웹환경에서 문서의 자동화된 교환 및 검색을 위해서는 문서 자체에 그 문서의 의미를 설명하는 구조를 추가적으로 가지고 있어야 한다.

(그림 1)은 자동화된 웹문서 교환 및 검색을 위한 요소 - 문법, 구조 그리고 의미를 보여준다.



(그림 1) 웹문서 교환 및 검색을 위한 요소

† 준회원 : 고려대학교 컴퓨터학과 박사과정
 †† 준회원 : 고려대학교 컴퓨터학과 석사과정
 ††† 정회원 : 강릉대학교 컴퓨터학과 교수
 †††† 정회원 : 고려대학교 컴퓨터학과 교수

웹문서에 있어서 의미(Semantics)는 문서를 기술하는 메타데이터로 표현된다. 더블린 코어[3]나 GILS 코어[4]와 같은 메타데이터 요소들은 문서 제목, 문서작성날짜 등 문서에 대한 핵심적인 기술 사항을 표준으로 정의하고 공통으로 사용하도록 한다. 구조(Structure)는 메타데이터 요소들을 웹문서에 수용하기 위한 틀로써, 기계-가독형(Machine-Understandable) 문서를 생성하여 자동화된 문서의 교환 및 검색을 이루는 바탕이 된다. 마지막으로 문법(Syntax)은 문서를 생성할 때, 상대방이 이해할 수 있는 형태로 가공하는 규칙이다. HTML과 XML은 이러한 규칙의 집합이다.

본 연구에서는 웹문서의 교환 및 검색을 위한 기본 프레임워크 중에서 문서의 의미 표현 방법 및 구조에 대해 현재 진행되고 있는 작업들을 소개하고 그 전망을 살펴본다.

서론에 이어 2장에서는 문서 설명을 위한 표준 메타데이터 요소들에 대해 설명하며, 3장에서는 메타데이터와 웹문서의 결합을 위한 모델들을 소개하며, 이들이 XML문법으로 구현되는 방법을 설명한다. 4장에서는 표준 데이터 요소들을 각 도메인에서 활용하고 서비스하는 방안으로 데이터 레지스트리를 소개하고 레지스트리를 이용한 문서 교환 및 검색 프레임워크를 제시한다. 마지막으로 5장에서 결론을 맺는다.

2. 웹문서의 의미 표현

개별 조직의 정보 자원을 관리하고 외부 조직 간의 정보 호환성과 공유성을 높이기 위해 메타데이터의 역할에 대한 인식이 증가되고 있으며, 전자 상거래와 웹기반의 자원 검색 및 교환을 중심으로 메타데이터를 이용하는 방안이 활발하게 연구되고 있다.

이 장에서는 현재 웹문서 의미 기술 방법 중 가장 큰 비중을 차지하고 있는 더블린 코어 메타

데이터 요소와 GILS에서 정의하고 있는 메타데이터 요소를 설명한다.

2.1 더블린 코어(Dublin Core)

더블린 코어의 기본 목적은 네트워크 자원의 검색에 적합하면서도 작성이 간단한 핵심 데이터 요소의 정의이다[3]. 네트워크자원의 기술에 필요한 일련의 데이터요소를 규정하고, 이들 자원의 신속한 검색을 목적으로 1995년 OCLC(Online Computer Library Center)와 NCSA(National Center for Supercomputer Applications)가 더블린(Dublin)에서 개최된 워크숍에서 합의된 메타데이터 요소들을 더블린 코어라고 부른다.

더블린 코어는 MARC와 같은 기존의 메타데이터로는 네트워크자원을 표현하는데에는 구조의 경직성으로 인해 많은 비용과 시간이 소요되기 때문에 이를 대체할 수 있는 단순구조의 형식을 모색한 결과이다. 이에 따라 기본 데이터요소의 선정과 기술의 기준을 다음과 같이 제시하였다[3].

- 고유성: 자원의 본질적인 특성을 기술요소로 한다.
- 확장성: 규정된 필수 데이터요소 이외에 부차적인 내용이나 특성을 기술요소로 사용한다.
- 구문 독립성: 응용분야나 표현기법을 규정하지 않는다.
- 선택성: 각 요소의 수록여부를 강제하지 않는다.
- 반복성: 모든 기술요소를 반복 사용할 수 있다.
- 수정 가능성: 한정어를 사용하여 세부사항을 조정한다.

(그림 2)는 더블린 코어에서 정의하는 핵심 데이터 요소[5]들을 보여준다.

최근 네트워크 상의 웹문서에 대한 메타데이터 표준 모델로 정착되어 가고 있는 더블린 코어는 자원 기술의 단순성(Simplicity), 융통성(Flexibility),

· 제목(Title)	· 표현형식(Format)
· 저자(Author)	· 식별기호 (Identifier)
· 주제(Subject)	· 출처(Source)
· 기술(Description)	· 언어(Language)
· 발행자(Publisher)	· 관계(Relation)
· 기여자(Contributor)	· 범위(Coverage)
· 발행일(Date)	· 권리(Rights Management)
· 유형(Resource Type)	

(그림 2) 더블린 코어 데이터 요소

의미적 상호운용성(Semantic Interoperability)을 장점으로 하여 네트워크 자원의 검색에 적합하면서도 레코드 작성이 용이하고, 또 여러 분야에서 공통으로 사용될 수 있는 간단한 핵심 데이터 요소로 자리잡고 있다.

독일 수학협회는 웹 탐색엔진인 하베스트(HARVEST)를 통해 출판전 배포기사를 웹을 통해 접근할 수 있는 방법을 더블린 코어 요소를 기반으로 정의된 17개의 DMVPreprint-Core 요소를 사용하여 제시하고 있다.

또한, EdNA(Education Network Australia)[6]는 호주의 교육관련기관들이 정보기술을 응용하기 위해 설립한 교육망으로서 EdNA 데이터베이스 상에 있는 자원이 다른 목록 서비스에 의해 탐색될 수 있도록 자원에 대한 상세한 기술을 위해 더블린 코어 기반의 메타데이터 요소를 개발하였다. 더블린 코어에 정의되어 있는 15개 요소 중 Relation, Contributor, Source의 3개 요소를 제외한

12개 요소와, EdNA에서 필요로 하는 8개 요소를 추가하여 사용한다.

2.2 GILS의 핵심 데이터요소(GILS Core)

GILS(Global Information Locator Service)는 미국에서 정부부처 및 연방기관 정보에 대해 일반 대중들의 접근을 가능하게 하고 인터넷과 같은 개방 시스템 환경에서 소재 안내 자료의 검색을 위해 제안되었다.

GILS는 정보를 탐색하기 위하여 직접적으로 또는 중개자를 통해 사용되는 분산화된 위치들과 관련된 정보서비스들의 집합으로 물리적, 지리적으로 분리된 다른 장소의 정보 자원에 대하여 분산 접근을 가능하게 한다. 이는 분산 조직화된 컴퓨터 네트워크를 통하여 물리적, 지리적으로 분리된 다른 장소의 정보 자원에 대하여 지시자(locator) 역할을 하는 메타데이터를 통하여 정보 자원의 식별, 소재과역, 접근과 정보 획득을 용이하게 한다. 메타데이터 역할을 하는 지시자란 원격지 정보자원을 인식하고 그 정보소스가 제공가능한 정보를 표현하며, 그 정보를 획득하는 것을 돕는 정보를 모두 의미한다[7].

GILS 표준 레코드 구성하고 있는 GILS 코어는 미국 연방정부에서 관리하는 소재 지시 레코드의 세트로 GILS의 핵심 요소(Core Element) 규칙에 따른 것이고 네트워크 상에서 접근이 가능하다. 정보를 배포하는 각 기관은 GILS내에서 자관의

· 제목(Title)	· 대리프로그램(Agency Program)	· 미통제용어(Subject terms uncontrolled)
· 원저자(Originator)	· 공간적 영역(Spatial Domain)	· 통제색인(Controlled Subject Index)
· 기여자(Contributor)	· 시간 영역(Time Period)	· 상호참조시형(Cross Reference)
· 발행일(Date of Publication)	· 유용성(Availability)	· 스케줄 번호(Schedule Number)
· 발행지(Place of Publication)	· 날짜의 출처(Sources of Date)	· 통제기호(Control Identifier)
· 언어(Language of Resource)	· 방법론(Methodology)	· 최종 갱신일(data of last modification)
· 초록(Abstract)	· 접근제한사항(Access Constraints)	· 레코드 언어(Language of Record)
· 목적(Purpose)	· 이용자제한사항(Use Constraints)	· 원시통제기호(Original Control Identifier)
· 접촉점(Point of Contact)	· 부수정보(Supplemental Information)	· 레코드 출처(Date of Last Modification)

(그림 3) GILS의 핵심 데이터 요소(Core Data Element)

레코드에 대한 조합과 유지에 대한 책임을 진다. GILS 코어가 보유하고 있는 소재 지시 정보에 대한 서비스는 미국 정부가 관리하며 무료로 제공된다. GILS 핵심 데이터 요소[8]를 (그림 3)에 정리하였다.

현재 시범적으로 웹을 통해 서비스되고 있는 GILS의 각 선진국 사례를 분석해 보면 원래 GILS가 목표했던 정부 차원의 정보소계 안내 서비스에 대한 비전을 달성하지 못하고 있다. 이는 각 기관마다 GILS가 서로 다른 방식으로 구현되고 있으며 각기 다른 유틸리티를 사용하고 있어, 대다수 기관의 GILS가 서로 유기적이고 통합적인 형태로 서비스되지 못하기 때문이다.

3. 웹문서 설명을 위한 메타데이터구조

3장에서는 웹문서를 표준 메타데이터를 웹문서에 수용하기 위한 구조들을 기본 개념과 XML을 이용한 구현을 중심으로 설명한다.

3.1 RDF(Resource Description Framework)

W3C의 주도로 개발된 RDF(Resource Description Framework : 자원 기술 구조)는 구조화된 메타데이터를 표현하고, 교환하며 그리고 재 사용하기 위한 기반구조(Infrastructure)다[9]. 이 기반 구조는 메타데이터의 의미, 문법 그리고 구조에 대한 공통의 규약을 지원하는 메카니즘의 설계를 통해 메타데이터의 상호운영성을 가능하게 한다[10].

RDF는 메타데이터를 처리하고 교환하는 공통의 문법으로 XML을 이용한다. XML의 특징을 그대로 사용하기 때문에, RDF는 의미에 대한 명확한 표현을 지원하는 구조를 가지며, 표준화된 메타데이터에 대한 일관된 표현, 교환, 처리가 가능하다.

(1) RDF 데이터 모델

RDF는 각각의 자원 기술 조직이 정의한 어휘 집합(Vocabulary)을 그대로 사용한다. 그러나, 사

용되는 도메인에 따라 그 의미가 달라질 수 있으므로, 어휘의 출처까지 조합하여 사용한다. 예를 들어, (그림 4)는 “Document 1”의 저자가 “John Smith”라는 것을 더블린 코어에서 정의한 어휘 집합인 “Creator”를 이용해서 나타낸다.



(그림 4) RDF 데이터 모델

(2) XML을 이용한 RDF의 구현 예

(그림 4)를 XML 형식을 이용하여 나타내면 (그림 5)와 같다. <RDF:RDF>는 단순한 래퍼로서, XML문서 안에서의 RDF 영역을 표시하는 역할을 한다. <RDF:Description>은 RDF안의 설명이 대응되는 URI(Uniform Resource Indicator) 문서, “Document-1”에 대한 설명임을 나타낸다. <DC:Creator>는 더블린 코어에서 정의하는 “Creator”의 의미로 “Document-1”의 저자로 “John Smith”라는 값을 갖음을 나타낸다.

```
<?xml:namespace ns="http://www.w3.org/RDF/RDF/" prefix="RDF" ?>
<?xml:namespace ns="http://purl.oclc.org/DC/" prefix="DC" ?>
<RDF:RDF>
  <RDF:Description RDF:HREF="Document-1">
    <DC:Creator>John Smith</DC:Creator>
  </RDF:Description>
</RDF:RDF>
```

(그림 5) XML을 이용한 RDF 구현

RDF가 지원하는 규약은 분산된 메타데이터 요소 집합 사이의 상호운영성을 가능하게 한다. 이러한 규약은 단순하지만 강력한 데이터 모델에 근거하여 표현된 의미(Semantics)를 위한 표준을 수용한다. RDF는 또한 사람이 읽기 쉽고, 기계가 처리하기 쉬

운 단어들을 출간하는 방법을 제공한다. 이러한 단어들은 특성이나 메타데이터 요소와 같은 집합들로 자원 기술 조직에 의해 정의된다. 단어들의 정의나 표준화로 인해 재사용을 고무시키고, 다른 정보 조직들 사이의 의미를 확장할 수 있다.

실제로, 더블린 코어는 RDF를 받아들여기로 했으며[11], 교육 자료에 대한 접근을 제공하기 위해 설계된 Educom의 교육 메타데이터 시스템(IMS)[12]은 더블린 코어와 관련된 구조를 받아들이고 이를 확장하여 특정 도메인에 적합하도록 만들었다. RDF는 중심이 되는 하나의 레지스트리가 없이 여러개의 도메인 별로 레지스트리가 분산되어 존재할 때 이들의 의미를 유일하게 표현할 수 있는 기반 구조를 제공한다.

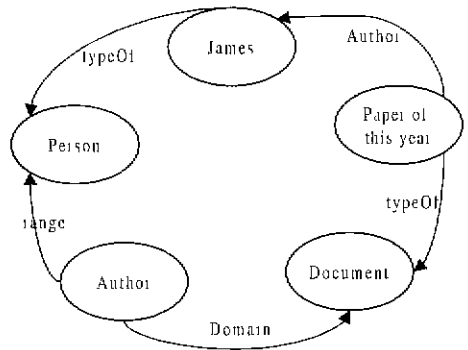
3.2 Meta Content Format(MCF)

차세대 언어로 XML(eXtensible Markup Language)이 선호되면서 푸시 기술(Push Technology)에 대한 적용이 증가하고 있다. 푸시 기술을 적용하기 위해서는 웹문서의 내용을 자동으로 분류하고 인식해야 한다. 이러한 푸시 기술을 지원하기 위해 넷스케이프 사에서 정의한 구조가 MCF[13]이다.

(1) MCF의 논리적 구조

MCF는 객체에 특성을 붙임으로써 문서에 대한 정보를 제공한다. 객체는 주로 웹페이지를 대상으로 하고, 특성은 그 웹페이지에 대한 설명 - 예를 들어, 크기, 제공자, 날짜 등 -을 제공하는데 사용된다. 이러한 객체와 특성을 MCF에서는 DLG(Directed Linked Graph)로 표현한다. (그림 6)은 DLG로 표현된 MCF의 논리적 구조를 보여준다.

(그림 6)에서 노드는 웹 페이지, 이미지, 주제별 범위, 채널, 사이트 등을 나타내며, 사람이나 장소, 사건과 같은 실세계의 객체를 표현하기도 한다. 호는 웹 페이지나 주제별 범위의 크기나 최종 수정일 같은 속성을 나타내며 또한 다른 객체와의 관계 즉, 하이퍼링크, 저작의 근원 등도 나타낸다.



(그림 6) DLG로 표현된 MCF의 논리적 구조

웹문서를 나타내는 “Paper of this year” 노드는 “Document” 노드로 가는 호가 있으며, 레이블은 “typeOf” 이다. 즉, 문서 “Paper of this year”의 타입이 “Document”임을 나타낸다. 또, 문서 “Paper of this year”는 “James”노드로 연결되어 있으며, 레이블은 “Author”이다. 이는 “Author”라는 특성 유형의 값이 “James”임을 나타낸다. “Author”노드는 “Person”과 “Document”로의 호가 있으며, 각각 “range”, “Domain”의 레이블을 가진다. “range”는 “Author”객체가 가질 수 있는 값의 범위를 말하며, “Domain”은 “Author” 객체가 존재하는 범위를 나타낸다.

(2) XML을 이용한 MCF 구현

(그림 6)은 XML을 이용하여 (그림 7)과 같이 표현된다. MCF를 기술하는 텍스트는 <XML-MCF>와 </XML-MCF>로 태그된 블록 안에 들어간다.

```

<XML-MCF>
  <Document id="Paper of this year">
    <author unit="James"/>
  </Document>
  <Person id="James">
  </Person>
  <PropertyType id="Author">
    <domain unit="Document"/>
    <range unit="Person"/>
  </PropertyType>
  <PropertyType id="Document">
  </PropertyType>
  .....
</XML-MCF>
    
```

(그림 7) XML을 이용한 MCF표현

MCF는 아직 그 구체적인 표준이 확정된 상태는 아니며, 구현된 사례 역시 그 가능성을 시험해 보기 위한 프로토타입 정도가 대부분이다. 대표적인 구현 사례로는 Apple사의 Hotsauce가 있다.

MCF는 웹상의 내용을 표현하는 방법으로 기존의 메타 태그를 사용하는 것보다 복잡하다는 단점이 있는 반면, 웹 문서를 보다 정확하고 구조적으로 설명할 수 있는 장점이 있다.

3.3 워릭 프레임워크(Warick Framework)

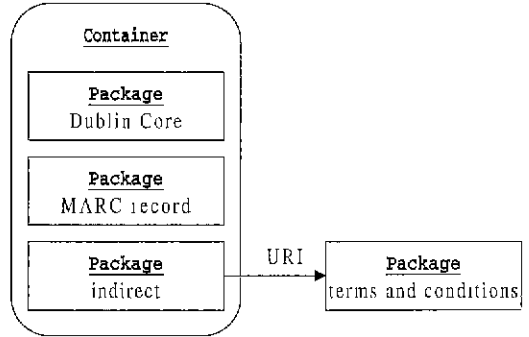
더블린 코어에 정의된 메타데이터는 네트워크 자원 표현의 필수적인 요소로서, 다양한 웹 문서의 제작자나 이용자에게 쉽게 이용될 수 있다. 하지만, 더블린 코어에는 이용에 대한 현실적인 방안이 포함되어있지 않으므로, 더블린 코어를 보다 견고하고 상호 운영성을 증대시키는 방안이 제기되었는데, 이것이 워릭 프레임워크[14]이다.

(1) 워릭 프레임워크의 구조

워릭 프레임워크는 더블린 코어의 입력 및 교환을 위한 구체적 구문으로, 개별적인 이용자와 타 기관의 메타데이터 집합과의 개별적인 영역을 구분하여 제공하는 컨테이너 구조(Container Architecture)[14]를 말한다. 컨테이너는 임시적이거나 영속적이 성질을 갖는데, 영속적인 형태에서 컨테이너는 저장소, 클라이언트와 에이전트 사이에 이동 객체로 존재한다. 하나 이상의 서버에 저장되어 서버 사이에 URI를 사용하여 접근, 이용할 수 있다. 컨테이너 내에서 메타데이터 집합은 MARC나 더블린 코어 레코드 등을 포함하는 실제 메타데이터를 포함하는 패키지(Package)로 표현된다.

(그림 8)은 워릭 프레임워크의 단순한 컨테이너 예이다. 이 그림에서 컨테이너는 세 개의 논리적인 패키지를 포함한다. 더블린 코어 레코드와 MARC 레코드의 한 쌍의 패키지를 포함하고 있고, 세번째 패키지는 URI를 통해 간접적으로 참조된다. 접근조건 및 규정 메타데이터를 위한 구

문은 아직 정의되지 않은 상태이다.



(그림 8) 워릭 프레임워크 구조

(2) 워릭 프레임워크의 구현

(그림 9)에서 보듯이 워릭 프레임워크는 메타데이터 요소 집합을 패키지로 보며, 패키지의 집합이 컨테이너이다. 따라서 컨테이너를 나타낼 수 있는 태그인 <container>와 패키지를 나타내는 태그 <package>, 그리고 패키지 내에서 각 메타데이터 요소를 표현할 수 있는 태그인 <metadata>를 이용하여 표현한다.

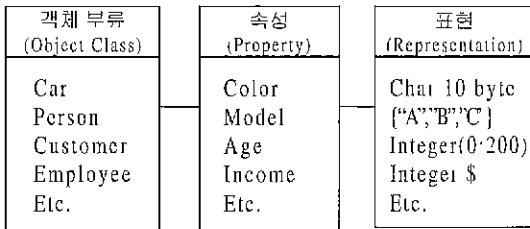
```
<container name="example">
  <package name="Dublin Core">
    <metadata name="title">Sample Document
                                     </metadata>
    <metadata name="author">Bob Dole
    </metadata></package>
  <package name="MARC" notation="RFC-822">
    <metadata name="from">daniel@acl.lanl.gov
    </metadata>
    <metadata name="subject">Metadata Tagging
    schemes </metadata></package>
</container>
```

(그림 9) 워릭 프레임워크의 XML구현 예

4. 데이터 레지스트리 중심의 웹 문서 교환 및 검색

4.1 데이터 요소와 데이터 레지스트리

환경, 교육, 전자상거래, 디지털 라이브러리, 전자정부 등 표준 메타데이터 요소가 실제 사용되어지는 분야별로 확장되어 정의될 때, 이들 공통 메타데이터들을 데이터 요소[15]라고 하며, ISO/IEC 11179에서 구조와 속성을 정의한다. (그림10)는 ISO/IEC 11179에서 정의한 데이터 요소의 구조를 보여준다.



(그림 10) 데이터 요소의 구조

ISO/IEC 11179에서 규정한 데이터 요소는 객체 부류(Object Class), 속성(Property), 표현(Representation)의 3부분으로 구성된다.

객체 부류(Object Class)는 자동차, 사람, 고객, 주문, 사원 등과 같이 우리가 수집하거나 저장하려는 데이터를 지칭한다. 속성(Property)은 색, 모델, 성, 나이, 수입, 주소, 가격 등과 같이 사물을 구별하거나 설명하는데 사용되는 특성이다. 표현(Representation)은 해당 데이터 요소가 가지는 값에 대한 도메인 및 표현 형태를 기술한다. 각 데이터 요소는 유일한 식별자(Identifier)를 가지며, 이 식별자는 데이터 레지스트리를 이용하는 모든 사용자 및 프로그램 사이에서 유일하다. 식별자 외에 데이터의 정의, 이름, 문맥, 표현형태 등이 기술된다.

(표 1)은 데이터요소의 예이다. (그림 10)의 객체 부류의 각 속성이 하나의 데이터 요소로 대응된다. 데이터 레지스트리는 이러한 데이터 요소를 등록하고 인증하며 사용자에게 제공한다.

(표 1) 데이터요소의 예

Identifier	DE Name	Representation
Emp001	Employee Name	Char 10 byte
Emp002	Employee Age	Integer(0:200)
Emp003	Employee Income	Integer \$
...
Car001	Car Name	Char 20 byte
Car002	Car Color	RGB color
Car003	Car Size	{소형,중형,대형}
...	..	
Cus001	Customer Name	Char 10 byte
Cus002	Customer Address	Char 30 byte

각 데이터 요소는 유일한 식별자(Identifier)를 가지며, 이 식별자는 데이터 레지스트리를 이용하는 모든 사용자 및 프로그램 사이에서 유일하다. 식별자 외에 데이터의 정의, 이름, 문맥, 표현형태 등이 기술된다.

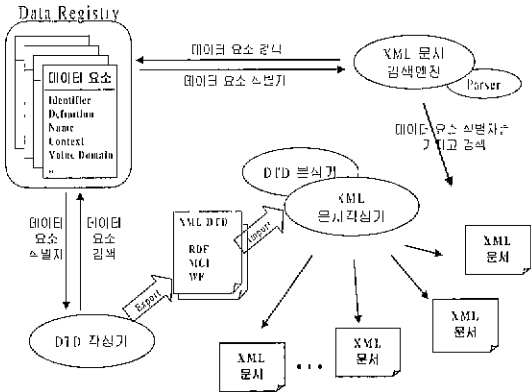
데이터 레지스트리는 데이터의 특성에 대한 사실들을 저장하는 장소이다. 데이터 레지스트리는 시스템 또는 조직 사이의 데이터 공유를 지원하며, 공유 데이터의 사용자로 하여금 데이터의 의미, 표현, 식별 등에 대한 일반적인 이해를 가능하게 한다. X3.285에서 데이터 레지스트리의 구조를 정의하고 있으며, X3.285를 기반으로 ISO/IEC 11179에서도 데이터 레지스트리의 기능 및 명세를 명시한다.

4.2 레지스트리 중심의 웹문서 교환 및 검색

본 연구에서 제시하는 XML문서 교환 및 검색 환경은 XML문서를 처리하는 시스템들 사이에서 문서를 이루는 태그의 의미를 레지스트리를 이용하여 공유하도록 함으로써, 문서의 구조뿐만 아니라 그 내용을 시스템이 자동적으로 인식할 수 있으며, 자동화된 문서 교환, 정확한 문서 검색을 지원한다[16].

전체적인 구조는 (그림11)과 같다. 데이터레지

스트리를 중심으로 ‘DTD 작성기’, ‘DTD 분석기’, ‘XML 문서 작성기’, ‘XML 문서 검색 엔진’으로 구성된다.



(그림 11) XML 문서 교환 및 검색 환경

에 대한 검색 결과를 얻을 수 있다.

즉, 제시된 환경에서는 XML문서마다 태그에 대한 정확한 의미를 데이터 레지스트리에 등록된 유일한 식별자를 사용하여 명시해 줌으로써, 태그의 형태에 무관하게 의미의 교환이 가능하며, 태그의 의미에 따른 문서의 검색이 가능하다.

```
<!DOCTYPE CAR SYSTEM Car.dtd >
<Car>
  <Nm ID="Car001"> 벤츠 </Nm>
  <Color ID="Car002"> 흰색 </Color>
  <Size ID="Car003"> 대형 </Size>
</Car>
```

(그림 12) 레지스트리에 기반한 XML 문서의 예

DTD작성기[17]는 XML DTD에서 정의하는 태그 및 속성을 데이터 레지스트리에 등록된 데이터 요소를 참조하여 작성하도록 도와준다. 작성된 DTD는 XML문서에 대한 일종의 스키마 역할을 하며 RDF, MCF, WF 등과 같은 메타데이터 구조를 표현할 수 있는 기반이 된다.

XML 문서 작성기는 DTD 분석기와 함께 작성된 DTD를 읽어들여 사용자가 DTD에 맞는 XML 문서를 작성할 수 있도록 도와준다.

(그림12)은 (표 1)의 데이터 요소에 기반하여 작성된, 자동차를 설명하는 XML문서의 예이다. 문서를 구성하는 태그마다 데이터 레지스트리에서 가져온 데이터 요소 식별자를 붙여 줌으로써 태그<Nm>은 자동차의 이름, <Color>는 색상, 그리고 <Size>는 크기를 나타낸다는 의미를 공유할 수 있다.

마지막으로, XML문서 검색도구는 제시한 환경에서 유통되는 XML문서에 대해 검색을 수행한다. 키워드 전문 검색과 함께, 문서를 구성하는 태그 및 속성에 대한 의미를 시스템이 인식할 수 있으므로, 사용자가 원하는 정확한 개념 및 문맥

5. 결 론

본 연구에서는 차세대 웹상에서의 자동화된 문서 교환 및 검색에 관한 노력에 대해 살펴 보았다. 문서 설명을 위한 메타데이터 요소를 표준화하는 노력으로 더블린 코어와 GILS 코어를 소개 하였으며, 차세대 웹문서 표준인 XML문서에 표준 메타데이터 요소를 수용할 수 있는 구조로 RDF, MCF, WF의 접근 방법을 XML을 이용한 구현 중점을 두어 설명하였다. 또한, 메타데이터를 비롯한 데이터 요소의 구체적인 서비스 방법으로 ISO/IEC 11179에서 정의한 데이터 레지스트리를 소개하고, 이를 이용한 문서교환 및 검색환경을 제시하였다.

웹 상에서의 문서 및 데이터의 의미 공유로 인한 자동화된 교환 및 검색은 개방환경에서의 시스템 통합의 핵심적인 사항이 될 것이며, 앞으로 많은 연구가 필요하다.

참고문헌

[1] D. Connolly and J. Bosak, "Extensible Markup

Language (XML)," 1997, <http://www.w3c.org/XML/>

[2] Rohit Khare, Adam Rifkin, "XML: A door to Automated Web Applications," IEEE Internet Computing, pp.78-87, July & August 1997

[3] "The Dublin Core Metadata," http://purl.org/metadata/dublin_core

[4] "Guidelines for the Preparation of GILS CoreEntries," <http://www.dtic.mil/gils/documents/naradoc/>

[5] "Description of the Dublic Core Elements," http://purl.oclc.org/metadata/dublin_core_elements

[6] "Metadata Description for dublin Core 5 Workshop," <http://www.otfe.vic.gov.au/edna/dc5edna.htm>

[7] "GILS (Global Information Locator Service/ Government Information Locator Service)," <http://www.gils.net/>

[8] "Guidelines for the Preparation of GILS Core Entries," Defense Technical Information Center, 1998, (<http://www.dtic.mil/gils/documents/naradoc/>)

[9] Ora Lassila, "Introduction to RDF Metadata," W3C NOTE 1997-11-13, <http://www.w3.org/TR/NOTE-rdf-simple-intro>

[10] Eric Miller, "An Introduction to the Resource Description Framework," D-Lib Magazine, May 1998, <http://www.dlib.org/dlib/may98/miller/05miller.html>

[11] "Dublin Core and Web MetaData Standards Converge in Helsinki", OCLC News Release, November 1997, <http://www.oclc.org/oclc/press/971107a.htm>

[12] "Instructional Management System," Project Website, April 1999, <http://www.imsproject.org/>

[13] R.V. Guha and Tim Bray, "Meta Content Framework using XML," June 6, 1997, <http://www.w3.org/TR/NOTE-MCF-XML/>

[14] Carl Lagoze, "The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata," D-Lib Magazine, June/August, 1996, <http://www.dlib.org/dlib/july96/lagoze/07lagoze.html>

[15] "Information technology - Specification and standardization of data element," ISO/IEC 11179-1 Final Committee Draft, June, 1998

[16] 나홍석, 채진석, 백두권, "데이터 레지스트리에 기반한 XML문서 공유 환경," 한국정보과학회 추계학술발표논문집, 25권 2호, pp.650-652, 1998

[17] 채진석, 나홍석, 백두권, "데이터 레지스트리를 이용한 자동화된 XML DTD 생성 방법," 한국정보과학회 춘계학술발표논문집, 26권 1호, pp. 9-11, 1999

나 홍 석



1994년 고려대학교 컴퓨터학과 (이학사)
 1996년 고려대학교 컴퓨터학과 (이학석사)
 1998년 고려대학교 컴퓨터학과 박사과정 수료

1996년-현재 고려대학교 기초과학연구소 연구원
 관심분야 : 데이터 공학, 데이터 레지스트리, 디지털 라이브러리, XML, 정보 검색

채 진 석



1998년 고려대학교 컴퓨터학과 (이학사)
 1998년-현재 고려대학교 컴퓨터학과 석사과정
 관심분야 : 데이터 레지스트리, XML, ERP, 객체 지향 모델링



김창화

- 1985년 고려대학교 수학교육과 (이학사)
- 1987년 고려대학교 전산학 전공 (이학석사)
- 1990년 고려대학교 전산학 전공 (이학박사)

1994년-1995년 캐나다 토론토 대학 Enterprise Integration 연구소 Post-Doc.

1998년-현재 한국정보처리학회지 편집위원

1989년-현재 강릉대학교 컴퓨터과학과 전임강사, 조교수, 부교수

관심분야 · 인터넷 정보검색, 멀티미디어 데이터베이스, 분산시스템, 멀티 에이전트



백두권

- 1974년 고려대학교 수학과 (이학 학사)
- 1977년 고려대학교 산업공학과 (공학석사)
- 1983년 Wayne State Univ. 전산학과 (공학석사)

1985년 Wayne State Univ. 전산학과(공학박사)

1990년-1991년 미국 Arizona 대학 객원교수

1996년-1997년 고려대학교 컴퓨터과학기술연구소 소장

1997년-1998년 고려대학교 정보전산원 원장

1986년-현재 고려대학교 컴퓨터학과 교수

1991년-현재 ISO/IEC JTC1/SC32 국내위원회 위원장

관심분야 : 데이터 공학, 데이터베이스, 소프트웨어 아키텍처, 데이터 웨어하우스, 데이터 마이닝, 디지털 라이브러리