

메타정보 인터페이스를 이용한 이질 구조 분석 XML 문서 통합 검색

(Integrated Information Retrieval with Metadata Interface for Heterogeneous Distributed XML Documents)

류성준[†] 황재문^{**} 김태훈^{***} 남영광^{****}

(Sung-Joon Ryu) (Jae-Moon Hwang) (Tae-Hoon Kim) (Young-Kwang Nam)

요약 본 논문은 구조적, 의미론적 이질성을 가진 분산 XML 문서의 통합 검색을 위해 반자동으로 생성된 인터페이스를 통해 각 지역 문서에 대한 질의를 생성하여 검색하는 방법을 제안한다.

본 시스템에서는 데이터 통합을 위한 메타데이터 인터페이스인 DDXMI(Distributed Documents XML Metadata Interface)를 정의하고, 분산 데이터에 대한 DTD를 입력으로 받아 사용자로 하여금 전역 DTD와 각 지역 DTD 간의 의미 차이를 극복하기 위한 사용자 인터페이스 생성 방법을 제안하였다. 전역 DTD와 지역 DTD의 특성을 고려하여 인덱스 매핑과 그에 필요한 함수 이름의 매핑 정보를 기반으로 DDXMI가 자동으로 생성된다. XML 질의 언어인 Quilt를 사용하여 생성된 DDXMI를 통해 각 지역 문서에 적합한 질의를 생성, 수행한다. 사용자는 검색 대상 문서의 스키마와 통합스키마의 구조를 잘 알고 있다고 가정하였다.

XML로 만들어진 석박사 논문, 논문지, 연구보고서에 대한 소규모, 중규모 전역 DTD를 만들어 실제로 질의를 생성하여 검색 결과를 검증할 수 있도록 하였다. 본 시스템은 JavaCC와 Java 서블릿을 이용하여 개발하였다.

키워드 : XML, 정보검색, 메타데이터

Abstract We propose an extremely light DDXMI approach for semi-automated integration of both structurally and semantically heterogeneous distributed XML documents. In the proposed prototype, a DDXMI(Distributed Documents XML Metadata Interface) is defined and a user interface generator is developed.

The prototype takes sources' DTDs as inputs and generates a friendly graphical user interface for the application users. The user can easily describe the semantic mapping between the integrated virtual database DTD and sources' DTDs through assigning index numbers and specifying associated function names so that the DDXMI based on the mappings is automatically generated. Quilt is selected as the XML query language which processes user queries according to the DDXMI. It is assumed that the application users know what they want from the different sources, that is, they have their own integrated database schema in their mind, and know the semantics of the involved XML databases.

A small-size global DTD and a mid-size global DTD are generated to verify the query generation and retrieval results with 3 XML document databases, that is, Master/Ph.D thesis, research reports, and journal databases. The system has been developed with JavaCC and Java Servlet.

Key words : XML, Information Retrieval, Metadata

· 본 논문은 2002년도 매지학술연구비와 2003년도 연세대학교 학술연구비 지원을 받아 이루어졌음

[†] 비회원 : 포스데이터 e-biz 연구소
sjryu@posdata.co.kr

^{**} 비회원 : LG 전자 정보통신사업본부
bh90210@lge.com

^{***} 비회원 : 연세대학교 전산학과
angeljin@hosu.yonsei.ac.kr

^{****} 종신회원 : 연세대학교 전산학과 교수
yknam@dragon.yonsei.ac.kr

논문접수 : 2003년 10월 15일
심사완료 : 2004년 9월 3일

1. 서론

급속한 컴퓨터 관련 기술의 성장으로 인하여 많은 양의 유용한 데이터가 생성되고 이러한 데이터들은 서로 다른 위치에 존재하는 상이한 사이트에 저장되어 있다. 데이터 통합을 필요로 하는 시스템을 구현하기 위해서는 이러한 서로 다른 사이트에 산재되어 있는 데이터들에 대한 통합과 분석이 요구된다. 이는 각 지역 데이터를 관리하는 각 지역 엔터티는 각각의 데이터에 대해 표현방법, 이름규칙, 의미론적인 해석 등에 있어서 독립적으로 조작되고 있기 때문이다. 이렇게 각 지역에 산재되어 있는 자료를 얻기 위해 필요한 데이터를 추출해서 하나의 공통 형식으로 바꾸는 방법은 적절한 해결책이 될 수 없다. [1]에 정의되어 있듯이 데이터 통합은 여러 종류의 문서를 입력으로 받아서 이러한 데이터들 사이의 연관 정보들을 기술하는 하나의 통합스키마를 구축하고, 이를 통해 각 분산 데이터에 대한 통합 검색을 지원하게 하는 것이다. [2]에서와 같이 도서정보, 지리, 환경, 생태학 등 다양한 분야의 도메인에 대한 표준화 작업이나, 임시방편의 표준 채택(예를 들어 Dublin Core [3]), 혹은 메타데이터 표준의 증가는 시스템적, 무문론적, 구조적 그리고 제한적이긴 하지만 의미론적 상호작용을 가능하게 하였다. 그러나 불행하게도 표준화 작업에 의한 데이터 통합을 통해 비교적 간단한 방법으로 이러한 기능들을 처리한다는 것은 아직 비현실적인 과제로만 남아 있는 상황이다. 결과적으로, 데이터 통합 과정은 많은 작업량과, 전문 기술을 요구하기 때문에 일반 응용 사용자가 혼자 힘으로 처리하기에는 매우 어려운 일이다.

데이터 통합을 하는데 있어서의 중요한 문제점은, 다른 출처에 있는 데이터들의 형식이 서로 다르고 변화한다는 것이다. 또한 이러한 데이터들은 서로 조화되지 않을 뿐더러 서로 상이한 의미를 내포할 수 있다. 예를 들어, 각 출판사의 도서정보 문서들은 서로 다른 가격의 단위를 사용 할 수도 있고, 저자 이름을 나타내기 위해서 서로 다른 형식을 사용 할 수도 있으며, 미국의 한 문서에서는 "CA"가 "California"를 나타내는데 사용 되는 반면, 캐나다의 한 문서에서는 "Canada"를 나타내는데 사용될 수도 있듯이, 같은 데이터가 다른 의미로 사용 될 수 있다. 이러한 이유로, 여러 사이트에 존재하는 분산 데이터들은 시스템적, 무문론적, 그리고 구조적인 이질성 외에도 의미론적 이질성을 가중시킨다. 데이터에 대한 무문론적인 표현이나 메타데이터만으로는 모든 가능한 데이터 통합을 위한 충분한 의미정보를 제공하지 못한다. 그렇기 때문에 데이터 통합을 연구하는 그룹 사이에서는 반자동적인 방법을 통해 메타데이터와 각 지

역 문서들과의 매핑 정보를 기술하여 통합 뷰를 생성하는 것이 일반적인 방법으로 인식되고 있다.

본 논문에서는 일반 응용 사용자가 지역 문서에서 필요한 정보가 무엇인지 알고 있고, 즉 사용자 자신의 통합 뷰를 미리 정해놓고, 지역 문서가 가지고 있는 의미론적 정보를 알고 있다고 가정하고 메타데이터와 지역 문서들 간의 매핑 정보를 기술 할 수 있는 손쉬운 사용자 인터페이스를 제공한다. 분산 데이터에 대한 데이터 통합을 위해 XMI[7]에 기반한 인터페이스인 DDXMI (Distributed Documents XML Metadata Interchange)를 제안하여 반자동적인 데이터 통합을 가능하게 하였다. 본 논문에서 제안한 DDXMI는 문서나 XML 문서의 이름, 테이블 열 이름과 XML 경로 정보, XML 문서의 요소, 속성들의 의미론적 정보를 포함한다. 시스템 프로토타입으로 사용되는 DDXMI DTD를 정의 하였으며, 사용자 인터페이스 생성기가 개발되었다. 사용자 인터페이스 생성기는 각 지역 DTD를 파싱하여 각 요소에 해당하는 경로 정보를 생성하며, 사용자에게 손쉬운 방법으로 각 요소 간 매핑 정보를 기술 할 수 있도록 사용자 인터페이스를 생성한다. 매핑 정보 기술은 전역 요소에 해당하는 지역 요소와의 인덱싱 방법과 필요한 특정 함수의 추가로 쉽게 이루어지며, 이때 사용되는 함수는 본 시스템에서 사용된 질의 언어인 Quilt[8]에서 제공하는 내장 함수 및 사용자 정의 함수를 이용한다. 매핑 정보 기술이 끝나면, 매핑 정보에 기반하여 동일한 인덱스 번호를 가지는 요소들을 수집하여 DDXMI가 자동으로 생성되고, 사용자에게 의해 지정된 전역 질의는 DDXMI를 참조하여 각 지역 문서에 적합한 지역 질의를 생성하게 된다.

본 논문에서 제안한 방법을 사용함으로써 분산 환경에서의 각 지역 문서간의 개략적인 이질성 문제를 해결할 수 있다. 또한, Quilt를 사용함으로써 질의 최적화 및 질의 실행 문제를 고려할 필요가 없기 때문에 전체 시스템의 크기가 상당히 경량인 점도 특징으로 들 수 있다. 또 다른 주목할 만한 특징은 유용성을 들 수 있다. 즉, 각 사용자마다 여러 곳의 지역 문서에 대해 자신의 응용에 알맞는 자신만의 전역 DTD로 지정하여 응용에 사용할 수 있다. 또한 다른 통합시스템에서와는 달리, 통합을 위한 새로운 도구나 언어 혹은 모델을 정의함이 없이, 어떠한 구조의 문서가 주어지더라도 사용자에게 새로운 것에 대한 습득과 교육에 대한 부담 없이 기존의 도구와 약간의 GUI 인터페이스를 사용하여 통합검색을 할 수 있다는 것이 본 논문에서 제시한 시스템의 가장 큰 장점이다. 이와 관련된 내용들은 3장에서 시스템의 구조를, 4장에서는 DDXMI 생성 방법을 서술하였으며, 5장에서는 도서정보를 통한 실험을 상세히 기술하

였다. 본론에 앞서 관련 연구 및 본 논문에서 사용한 방법과의 비교를 간단히 2장(관련연구)에서 논한다.

2. 관련 연구

데이터 통합을 위한 관련 데이터들의 이질성 문제를 다루고 있는 다양한 해결 방법이 개발되었고 현재도 연구되고 있다. 스키마적인 불일치 문제에 초점을 맞추고 있는 스키마 통합[9]과 스키마 매핑[10-12] 방법은 의미 상호교환 문제에 있어서 일부분만을 해결하고 있다.

국내의 연구로서, XMF[30]는 XMF-ML이라는 중재 언어를 이용하여 이질성을 극복하기 위한 규칙과 통합 스키마를 정의하고, 사용언어로서는 XQL을 기반으로 제구성한 XMF-QL을 사용하였는데, 이 경우는 XMF라는 새로운 규칙과 언어를 정의하고 다시 이를 처리할 수 있는 프로세서가 필요하다. ETRI에서 개발된 Data-Blender[29]는 통합스키마를 위하여 XquerySD라는 언어를 이용하여 스키마를 정의하기 때문에 GUI를 통한 인터페이스보다는 통합스키마의 작성이 어렵다.

국외의 연구로서, 두개의 서로 다른 데이터 표현에 대해서 편리한 사용자 인터페이스를 통해 매핑 관계를 생성할 수 있는 스키마 매핑 툴인 Clio는 IBM사의 Almaden 연구소에서 개발되고 있다. Clio에 의해 생성된 질의어는 메타 질의 엔진인 Garlic에 의해서 실행되는데, 소스 데이터를 받아서 타겟 스키마에 맞게 변환시키는 역할을 한다[10,11]. 이와 관련된 언어 기반 접근 방법들은 데이터의 특징을 기술 할 수 있고, 스키마 변환을 위한 툴을 제공한다. 예를 들어 YAT 변환 언어 [13]과 [14,15]에서 제시한 일치규칙에 의한 방법들은 데이터의 특징들과 스키마간의 매핑을 가능하게 한다. [16, 17]에서 데이터 통합을 위해 제시한 방법인 MIX Mediator System은 상당히 좋은 성능을 가지고 있지만 매우 복잡한 방법을 사용하며, XSL[18], Quilt[8]를 사용하는 일부 솔루션들은 비교적 간단하게 XML 문서 간에 변환이 이루어지도록 하였다.

이러한 방법들에서 해결해야 할 가장 큰 문제점은 의미 문제를 해결 하는 것이다. 즉, 데이터 소스들의 내용 및 표현을 실세계의 개체와 개념에 연관시키는 일이다 [19]. 그러나 위에 언급했던 구분론적인 메타데이터에 근거한 접근 방법들로는 데이터들 간에 존재하는 모든 의미론적 상호운용을 보장하지 못한다. 이와 같은 맥락으로 온톨로지(Ontology) 및 그와 유사한 구조를 개발 하는데 많은 노력을 기울이고 있는 실정이며, 이러한 예로는 RDF[20], Knowledge Sharing Effort[21], Intelligent Integration of Information[22], Digital Library Initiative[23], 그리고 Knowledge-based Integration [24,25] 등을 들 수 있다. RDF는 컴퓨터가 처리할 수

있도록 데이터의 의미를 기술하는 표준을 제공하지만, 제약조건 및 규칙(예를 들어 클래스 정의)을 정의하는 방법을 제공하지 않기 때문에 사용자가 직접 이러한 정보들을 기술해 주어야 한다. XML 구문을 기반으로 하여 만들어진 F-Logic[24-26], Ontolingual[27], XOL[28]과 같은 온톨로지 언어는 데이터와 지식을 표현하는데 사용되고 있으며, W3C에서 재정한 RDF와 RDFS를 기반으로 하여 만들어진 OIL, DAML+OIL, OWL 등이 있다[31]. 온톨로지 시스템을 이용한 시스템의 통합은 온톨로지 시스템 자체가 방대할 뿐만이 아니라, 언제 어느 때이던지 사용자가 수정하여 사용하기가 어려우며, 온톨로지와 검색시스템을 다시 연계시켜야하는 문제점이 있다.

실제로 많은 응용 사용자들은 위에서 언급된 방법과 같은 복잡한 방법들의 사용보다는 임시방편의 방법을 통해 스스로 데이터 통합을 가능하게 하는 방법을 선호한다. 본 논문에서는 응용 사용자들에게 DDXMI를 통해 스스로 데이터 통합을 가능하게 하는 쉬운 방법을 제시한다. 특별한 데이터 통합을 요구하는 작업을 위해서 친숙한 사용자 인터페이스를 통해, 손쉽게 기술 할 수 있는 의미론적 메타데이터를 바탕으로 DDXMI가 자동으로 생성된다. 이는 입력으로 받은 여러 개의 각 지역 DTD를 기반으로 생성되며, 전체 시스템은 아주 간단한 구조를 가지므로 응용 사용자들이 손쉽게 조작할 수 있어 유용성과 의미론적 데이터 통합을 위한 충분한 기능을 제공한다.

3. 시스템 구조

DDXMI를 기반으로 한 분산 문서 시스템의 전체적인 구조는 그림 1과 같다. 모든 문서는 XML 형식으로 표현 가능하고, 문서를 위한 테이블 및 데이터 구조는 XML DTD에 저장된다고 가정하였으며, XML 질의 언어인 Quilt를 데이터 조작을 위한 언어로 사용하였다.

기본 개념은 사용자에 의해 생성된 질의어(이하 전역

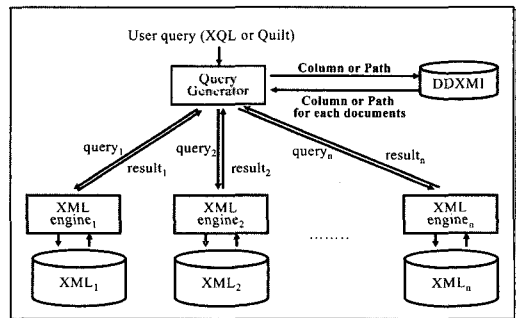


그림 1 시스템 구조

질의어)가 질의어 생성기에 의해 DDXMI에 저장되어 있는 매핑 정보를 이용해 각각의 XML 문서 형식에 적합한 질의어(이하 지역 질의어)로 변환하는 것이다. DDXMI는 각 지역 XML 파일에 적용되는 경로 정보와 각 지역 문서에 적용될 함수 정보를 가지고 있다. 전역 질의어 내에 존재하는 경로 정보는 질의어 생성기에 의해 파싱된 후, 각 요소에 해당되는 매핑 정보가 DDXMI 내에 있을 경우 각각의 지역 문서 형식(지역 DTD)에 해당되는 경로 정보로 대체 되거나, 매핑 정보가 없다면 널 값으로 변환된다. 질의어 생성기에 의해서 생성된 지역 질의어는 각각의 문서 엔진으로 보내진 후 질의 수행을 하게 되고 그에 해당하는 결과 값은 전역 질의어에 대한 결과가 된다. 물론 여기에는 많은 중복된 결과가 포함 될 수 있고, 또한 지역 질의어의 결과에 대한 조인 작업이 필요할 수 있다. 이러한 문제는 문서 엔진에서 처리가 가능할 것으로 생각된다.

4. Distributed Documents XML Interface (DDXMI)

4.1 DDXMI DTD

본 논문에서 제안한 프로토타입인 DDXMI 역시 XML 문서이다. 여기에는 문서간의 경로 관계에 관한 메타 정보를 포함하고 있으며, 각 지역 문서간의 의미론적, 구조적 상이성 문제를 처리하기 위한 함수 이름을 포함한다. 이러한 DDXMI를 위한 DTD는 그림 2와 같다.

```
<ELEMENT DDXMIA (DDXMI.header, DDXMI.isequivalent, documentspec)*>
<ELEMENT DDXMI.header (documentation,version,date,authorization)>
<ELEMENT documentation (#PCDATA)>
<ELEMENT version (#PCDATA)>
<ELEMENT date (#PCDATA)>
<ELEMENT authorization (#PCDATA)>
<ELEMENT DDXMI.isequivalent (source,destination)*>
<ELEMENT source (#PCDATA)>
<ELEMENT destination (#PCDATA)>
<ELEMENT documentspec (document, (elementname,operation)*)*>
<ELEMENT document (#PCDATA)>
<ELEMENT elementname (#PCDATA)>
<ELEMENT operation (#PCDATA)>
```

그림 2 DDXMI.DTD

“source” 요소는 본 시스템을 사용하기 원하는 사용자나 시스템 매니저에 의해서 생성된 전역 문서에 있는 각 요소의 이름이고, “destination” 요소는 각 “source” 요소에 해당하는 각 지역 문서에 존재하는 요소의 이름이다. 질의 생성기가 전역 질의어에서 “source” 요소에 해당하는 요소 이름을 찾아내면, 지역 질의어의 해당하는 경로가 “destination” 요소의 경로로 바뀌게 된다.

서로 다른 사이트에 존재하는 몇 개의 출판사 문서를 예를 들어보자. 전역 문서에 존재하는 “author” 요소는 각각의 지역 문서의 해당되는 요소 이름인 “author”,

“author-name”, “name”, “auth”로 표현된다. 이때 DDXMI 내에서 “author”에 해당하는 “source” 요소는 “destination” 요소로 “author”, “author-name”, “name”, “auth” 요소로 매칭 된다.

4.2 DDXMI 생성을 위한 GUI

각 지역 문서는 XML 형식으로 표현 할 수 있기 때문에, 각각의 XML 문서는 자신의 DTD 파일을 가진다. 이는 전역 DTD 내의 필드가 같은 의미를 가지는 각 지역 DTD 내의 필드와 연관될 수 있다는 것을 의미한다. 이러한 전역 DTD 내의 요소는 연관되는 각 지역 DTD 내의 요소들과 함께 DDXMI에 수집되어야 한다. 전역 DTD 내의 어떤 요소와도 연관될 수 없는 지역 DTD 내의 요소들은 DDXMI에 포함되지 않는다.

본 논문에서는 DTD 내에 단지 말단 요소만 사용된다고 가정하였다. 실제로, 모든 말단 요소들은 n-차원 트리로 표현 가능하다. 만일 전역 DTD 내의 각 노드를 지역 DTD 내의 노드와 1:1 매치시킬 수 있다면 DDXMI는 손쉽게 생성될 수 있다. 전역 DTD 트리내의 각 노드에 인덱스를 부여한다. 그리고 각각의 전역 DTD 노드에 대해서 지역 DTD 트리의 같은 의미를 가지는 노드와 같은 인덱스를 부여하게 되는데, 이는 같은 인덱스를 가지게 되는 노드는 같은 의미를 가진다는 것을 의미한다. 그 후, 같은 인덱스를 가지는 모든 노드들을 모아서 DDXMI의 “source” 요소와 “destination” 요소들은 자동으로 생성된다.

예를 들어, 그림 3은 하나의 지역 문서를 정의하고 있는 DTD 파일을 보여주고 있다. 이런 DTD 내에 존재하는 각 요소들은 DTD 파서로 부터 파싱되어 그림 4와 같은 형태로 사용자에게 보여지게 된다.

그림 4에서 사용자 입력을 요구하는 첫 번째 열에는 DDXMI 생성을 위한 인덱스 번호를 입력받게 되고, 같은 인덱스를 가지는 모든 노드들을 모아서 DDXMI의 “source” 요소와 “destination” 요소들은 자동으로 생성된다. 인덱스를 갖지 않는 노드도 생기게 되는데, 이는 데이터 통합을 요하는 본 시스템에서 필요하지 않은 데이터를 의미한다. 문서 이름 필드는 DTD 파일이 적용되는 XML 문서의 파일 이름이다. 사용자 입력을 요구하는 두 번째 열에는 필요한 의미론적 문제들을 해결

```
<ELEMENT bib (book* )>
<ELEMENT book (title, (author+ | editor+ ), publisher, price )>
<!ATTLIST book year CDATA #REQUIRED >
<ELEMENT author (last, first )>
<ELEMENT editor (last, first, affiliation )>
<ELEMENT title (#PCDATA )>
<ELEMENT last (#PCDATA )>
<ELEMENT first (#PCDATA )>
<ELEMENT affiliation (#PCDATA )>
<ELEMENT publisher (#PCDATA )>
<ELEMENT price (#PCDATA )>
```

그림 3 Book1.DTD 파일

Document Name	Index number	Function name
book.xml	1	
book	1	
@year	1.4	
title	1.3	
author	1.2	
last	1.2.1.2	
first	1.2.1.1	f1
editor	1.8	
last	1.8.2	
first	1.8.1	
affiliation	1.8.1	
publisher	1.6	
price	1.1	

그림 4 Book1.DTD를 위한 트리

하기 위한 함수 이름을 입력 받게 된다.

4.3 전역 DTD와 지역 DTD간의 매핑

위와 같이 GUI를 이용하여 인덱스 번호를 할당할 때 각 문서나 XML 문서는 같은 요소에 대해 서로 다른 구조를 가지게 되므로, DDXMI를 생성하는 과정에서 발생 할 수 있는 각 경우에 대해 올바른 처리를 가능하게 하는 방법이 제공되어야 한다.

4.3.1 1:1 매핑

전역 DTD와 지역 DTD 간의 요소가 일대일로 매핑이 되는 경우이다. 이 경우는 특별하게 고려할 사항이 없으므로 해당되는 지역 요소의 이름으로 매핑시킨다.

4.3.2 N:1 매핑

전역 DTD 내의 여러 원소들이 지역 DTD 내의 한 요소의 정보를 분리하여 가지고 있는 경우이다. DTD의 요소가 부가적인 기능을 요구하는 함수가 필요할 경우에는 그 함수 이름이 요소 이름에 추가 된다.

그림 5에서 "first_name"과 "last_name" 요소는 하나의 요소에 매핑 된다. 즉 Book3의 DTD 트리에서 "name" 요소는 전역 DTD의 두 요소인 "last_name"과 "first_name"에 동시에 매핑 된다. 이때 필요한 분리함수인 "fstring"과 "lstring"은 DDXMI 파일에 포함되고 지역질의 생성시에 질의생성기가 자동으로 해당되는 요소에 적용한다. 이러한 함수들은 질의 언어에서 제공되는 내장함수 혹은 사용자 정의함수로 사용할 수 있기 때문에 적용하기가 쉽다. 그림 6은 그림 5와 같은 형태로 매핑이 이루어 졌을 때 DDXMI 안에 생성되는 "source", "destination" 요소와 필요한 함수를 나타낸 것이다.

4.3.3 1:N 매핑

1:N 매핑은 전역 DTD 내의 한 요소가 지역 DTD 내의 여러 요소로 매핑이 되는 경우이다. 이 경우에는 특별히 필요한 함수는 없지만, 질의 변환 시, 전역 DTD 내의 한 요소에 해당하는 지역 DTD 내의 모든 요소들을 수집해서 경로 정보에 포함시켜야 한다. 예를 들어,

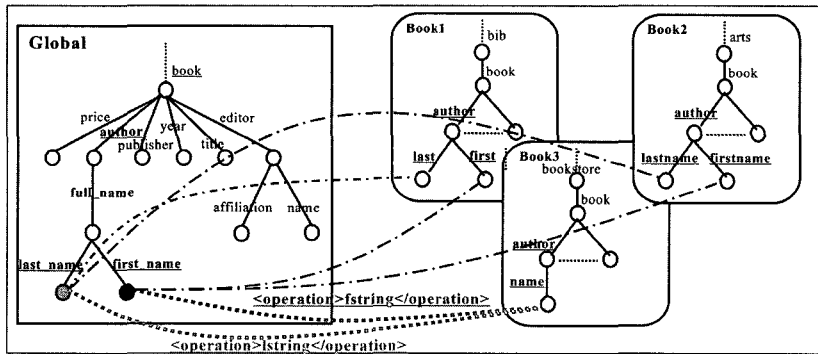


그림 5 N:1 매핑 예

```

<source>/book/author/full_name/first_name</source>
  <destination>/bookstore/book/author/name</destination>
<source>/book/author/full_name/last_name</source>
  <destination>/bookstore/book/author/name</destination>

<documentspec>
  <document>/book.xml</document>
  <elementname>/book/author/full_name/first_name</elementname>
    <operation>fstring</operation>
  <elementname>/book/author/full_name/last_name</elementname>
    <operation>lstring</operation>
  
```

그림 6 N:1 매핑 시 DDXMI 생성 예

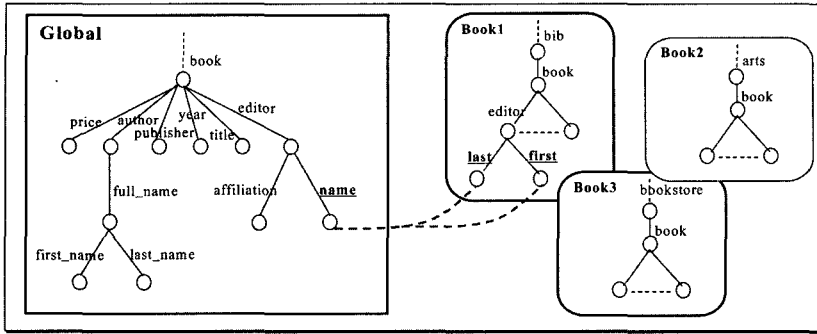


그림 7 1:N 매핑 예

```
<source>/book/editor/name</source>
<destination>/bib/book/editor/last,/bib/book/editor/first</destination>
```

그림 8 1:N 매핑 시 DDXMI 생성 예

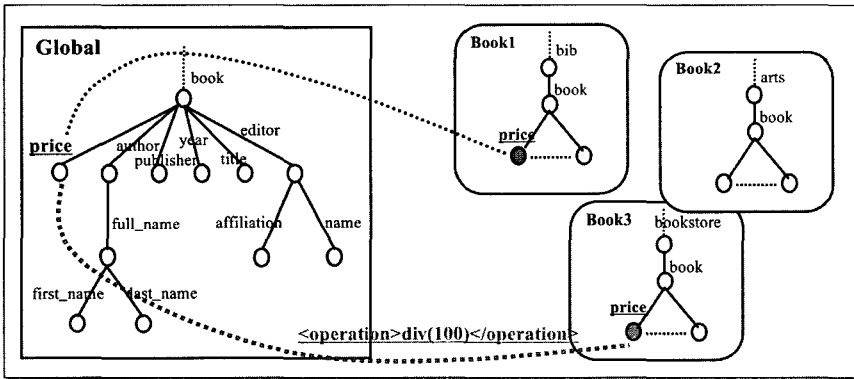


그림 9 의미정보를 포함하는 1:1 매핑

전역 DTD 내의 “full_name” 요소가 지역 DTD에서는 “last”, “first” 요소로 분리 되어 표현된 경우이다. 그림 7은 이러한 상황을 그림으로 표현한 것이고 그림 8은 DDXMI 안에 생성되는 “source”, “destination” 요소의 경로 정보를 나타낸 것이다.

4.3.4 의미정보를 포함하는 1:1 매핑

앞서도 언급했듯이 전역 DTD와 지역 DTD 간에 비록 같은 이름의 요소가 매핑 된다 하더라도 표현상 혹은 의미상의 이질성이 존재하는 경우를 고려해야 한다. 예를 들면, 그림 9에서 전역 DTD에는 통화 단위가 달려진 반면에, 한 지역 DTD에서 사용되는 통화 단위는 센트인 경우이다. 이러한 경우 매핑 정보 입력 시 전역 DTD 내의 요소와 동일한 의미를 가지도록 적절한 조치를 취해야 한다.

그림 9는 이러한 상황을 나타낸다. 질의가 파싱될 때, “price” 요소는 Book3를 위한 지역 질의를 생성 시

“price/div(100)”으로 교체 된다. 이는 전역 DTD와 지역 DTD 간의 의미 정보를 일치시키기 위함이다. 위의 네 가지 경우에 대한 매핑 방법을 요약하면 다음과 같이 표현될 수 있다.

4.4 개체(entity) 처리

XML의 DTD에는 일반 개체(general entity)와 파라메타 개체(parameter entity)의 두 종류가 있다. 일반 개체는 XML DTD에서는 사용되지 않기 때문에 여기서는 다루지 않는다. 파라메타 개체는 주로 반복해서 사용되는 DTD의 일부를 정의하여 사용자에게 편리성을 제공해 준다. 그러나 이 개체를 전역 DTD와 지역 DTD를 매핑하기 위해 모두 다 펼쳐서 파스트리로 나타내면 트리가 너무 커져서 색인을 할당하는데 문제가 있으며 메모리에 적재하기도 힘들다.

만약 지역 DTD도 개체가 정의되어 있어서 지역 개체와 전역 개체가 서로 대응될 경우, 한번만 전역 DTD

표 1 매핑 종류별 DDXMI에 포함될 내용 및 질의 생성 방법

매핑 종류	DDXMI에 포함될 내용		질의 생성
	전역 DTD	지역 DTD	
1: 1	요소이름	요소이름	요소이름만 대체
1: N	요소이름	N개의 요소이름과 이에 적용될 함수 이름 포함	LET, FOR, WHERE, RETURN 절에서 N개의 경로 생성
N: 1	N개의 요소 이름	요소이름, 전역 DTD의 N개의 요소에 적용될 함수이름 포함	LET, FOR, WHERE, RETURN 절에서 N개의 경로 생성
의미정보를 포함하는 1:1	요소이름	요소이름, 함수이름	지역질의에 포함된 요소에 함수이름 적용

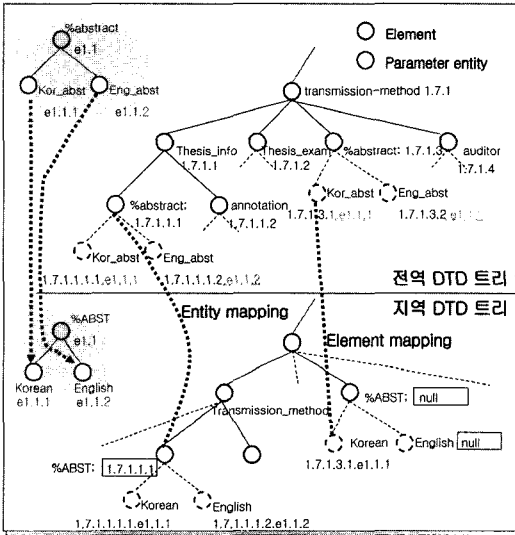


그림 10 개체 처리 방법에 관한 연구

와 지역 DTD를 매핑해 다른 모든 같은 이름의 개체는 일괄적으로 적용이 되도록 하면 사용자에게도 편리함을 줄 수 있을 것이다. 그러나 만약 사용자가 전역 개체를 지역 개체에 매핑하지 않고 일반 요소로 매핑하면 그 경우에만 개체 트리와 지역트리의 요소와 매핑하면 된

다. 예를 들어, 그림 10에서 전역 DTD의 “%abstract”와 지역 DTD의 “%ABST”가 매핑 될 경우 파스트리와 따로 분리되어 있는 개체끼리 매핑을 하면 파스트리에 나타나는 모든 같은 개체로 매핑이 된다. 하지만 전역 트리의 “1.7.1.3.1.e1.1.1”과 지역 트리의 “%ABST” 개체는, 밑에 있는 “Korean” 노드만 매핑 되고, “English” 노드는 매핑이 안 되어 색인번호가 “null”이 된다.

4.5 재귀적인 요소의 처리

DTD에서는 자연적으로 앞에서 나왔던 요소가 다시 자신의 요소로 나타나는 재귀적인 요소(recursive element)가 존재한다. 이러한 재귀적인 요소에 대해 아무런 제약을 가하지 않고 파스트리를 만들면 무한대의 트리가 만들어 진다. 이 문제에 대한 해결은, 매번 요소를 파싱하면서 그 요소가 자기의 조상 중에 나타난 적이 있으면 그 순간 더 이상의 파싱을 수행하지 않는다. 그림 11에서 전역 DTD의 “editor” 노드 밑에 다시 “book”이라는 요소가 나타난다. 이 경우 조상에 “book”이라는 요소가 존재하기 때문에 파싱을 멈춘다. 점선으로 표시된 부분에 있는 노드들은 다시 루트 밑에서 나타나기 때문에 더 이상 파싱을 할 필요가 없다. 만약에 전역 질의의 경로 “//book/editor/book/editor”에서 처럼 “book”과 “editor”가 반복이 되더라도 각 원소에 매핑되는 지역 DTD의 원소가 있기 때문에 전역 질의에 해

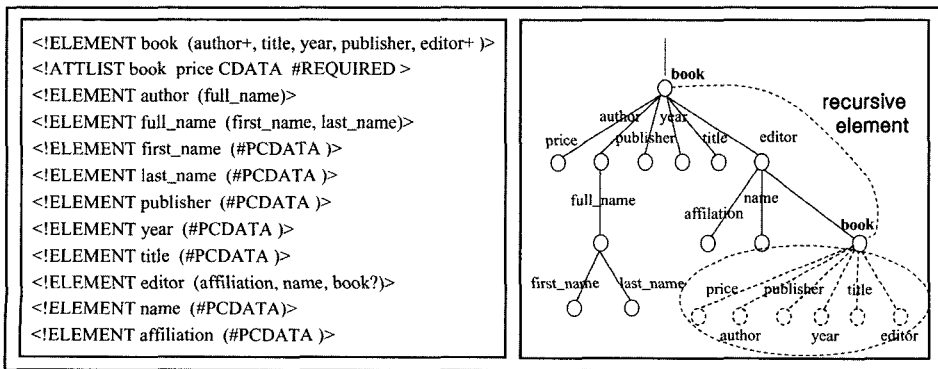


그림 11 재귀적인 요소를 가진 DTD 예

당되는 지역질의 생성하는데 문제가 없다.

4.6 질의 내에서의 경로 교체

본 논문에서 사용된 XML 질의 언어인 Quilt는 문서 내에서의 요소의 위치를 표현하기 위해 파일 시스템에서의 디렉토리 표현 형식과 유사한 경로 표현 형식을 사용하는데, 여기에는 두 종류의 경로가 존재한다. 즉, 상대 경로와 절대 경로이다. "/" 이후에 나오는 경로는 절대 경로를 의미하고, "/" 이후에 나오는 경로는 상대 경로를 의미한다.

DDXMI를 생성하는 과정에서 전역 요소들은 같은 인덱스를 가지는 지역 요소들을 가진다. 이 때, 인덱스가 부여된 두 노드 사이에 위치하는 몇몇의 노드는 인덱스가 생략될 수 있으며, 생략된 노드를 포함하는 하위 노드의 경로 정보는 생략된 노드를 포함하게 된다. 예를 들어, 그림 12에서 전역 DTD의 "price" 노드와 지역 DTD의 "price" 노드는 같은 인덱스를 가지지만 지역 DTD의 "price" 노드의 부모 노드인 "price_info" 노드에는 인덱스 번호가 부여되지 않았다. 이 때 지역 DTD를 위한 질의 생성시 "price" 요소는 "price_info" 요소를 포함하여 "/price_info/price"로 대체 된다.

5. 전역 DTD를 이용한 지역질의 생성 및 문서 검색 결과

본 논문에서는 구현을 위한 XML 질의 언어로서 Quilt를 사용하였으며, 질의 엔진으로는 University of Pennsylvania 에서 개발한 Kweelt를 사용하였다. 본 논문에서 제시한 방법에 대한 효용성을 입증하기 위해 그림 13과 같이 같은 정보를 가지면서 구조가 다른 도서정보에 대한 질의생성 예제와, 그림 14와 같이 완전히

다른 목적으로 생성된 문서이지만 공통이 되는 유용한 정보를 포함하고 있는 문서에 대한 질의 생성예제에 대한 결과를 표 2에서 보였다. 한국과학기술정보연구원서 제작한 석박사 논문용 DTD, 논문지 DTD, 연구보고서 DTD에 대한 정보를 검색하기 위해 저자와 제목을 포함하여 가장 기본적인 정보로 구성된 전역 표 3의 sample1.DTD와, 저자 및 제목을 포함하고 기타 약간의 상세정보를 포함하는 전역 표 4의 sample2.DTD를 만들어 실험하였다. 이들 두개의 전역 DTD에 대하여 세 개의 지역 DTD를 매핑시켜 sample1.DDXMI와 sample2.DDXMI를 생성하였으며, 생성된 DDXMI를 기반으로 각 전역 DTD에 대한 전역 질의를 만들고, 이 전역 질의에 대해 질의 생성기를 통하여 지역 질의를 생성하였으며, 생성된 지역 질의를 Kweelt 엔진을 통해 그 결과를 검증하였다. 사용자가 전역 DTD를 위한 질의어를 입력하면 생성된 DDXMI를 기반으로 각 지역 DTD에 알맞는 지역 질의어로 변환된다. 만일 DDXMI에 교체될 요소가 존재하지 않는다면 질의를 생성하지 않는다.

5.1 도서정보에 대한 지역질의 생성 예

그림 13과 같은 DTD 경로 정보로 구성되어 있는 도서관 전역 문서를 구축한다고 가정하자.

그림 14는 세 개의 서로 다른 DTD 파일에 대해서 노드에 대한 인덱스가 부여된 각각의 DTD 트리를 보여준다. 같은 의미를 가지는 요소에 대해서 같은 인덱스가 부여되었다. 자세히 살펴보면 인덱스의 순서가 서로 다르다는 것을 알 수 있다. 만일 전역 인덱스 파일에 "editor" 노드가 포함되지 않았다면 Book1.DTD 트리에서 "editor" 노드는 인덱스 번호가 부여되지 않았을 것이다. 또한 Book1.DTD 와 Book2.DTD는 "price" 노드

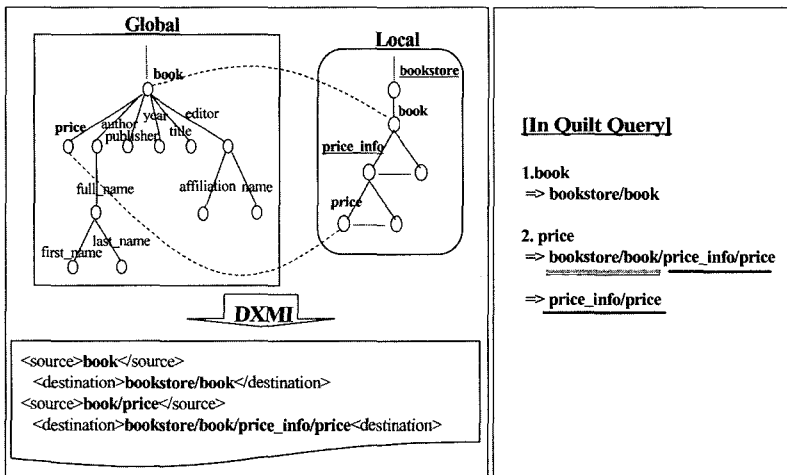


그림 12 질의 내에서의 경로 교체 예

표 2 전역 DTD [그림 13]과 지역 DTD [그림 14]에 대한 질의 생성 결과

전역 질의 (1:N 경우)	FOR \$a IN document("book.xml")//book//author RETURN <last>\$a/last_name</last> <first>\$a/first_name</first>
생성된 지역 질의	FOR \$a IN document("book1.xml") //book//author RETURN <last>\$a/last</last> <first>\$a/first</first>
	FOR \$a IN document("book2.xml") //book//author RETURN <last>\$a/lastname</last> <first>\$a/firstname</first>
	import split as UDF_split; FUNCTION firstStr(\$str) { split(" ",\$str)[1] } FUNCTION secondStr(\$str) { split(" ",\$str)[2] } FOR \$a IN document("book3.xml") //book//author RETURN <last> firstStr(\$a/name)</last> <first> secondStr(\$a/name)</first>
전역 질의 (N:1 경우)	FOR \$book IN document ("book.xml")//book[author/first_name="Franklin" AND author/last_name="Benjamin"] RETURN <book>\$book/title</book>
생성된 지역 질의	FOR \$book IN document("book1.xml") //book[author/first = "Franklin" AND author/last = "Benjamin"] RETURN <book>\$book/title</book>
	FOR \$book IN document("book2.xml") //book[author/firstname = "Franklin" AND author/lastname = "Benjamin"] RETURN <book>\$book/title</book>
	import split as UDF_split; FUNCTION secondStr(\$str) { split(" ",\$str)[2] } FUNCTION firstStr(\$str) { split(" ",\$str)[1] } FOR \$book IN document("book3.xml") //book[secondStr(author/name) = "Franklin" AND firstStr(author/name) = "Benjamin"] RETURN <book>\$book/title</book>
전역 질의 (1:1에서 함수포함)	LET \$title := document ("book.xml")//book[exists(@price)]/title RETURN <book>\$title</book>
생성된 지역 질의	FOR \$book IN document("book1.xml") //book[price .>. "8" AND author/first = "Melville" OR author/last = "Herman"] RETURN <book>\$book</book>
	생성된 질의 없음
	import split as UDF_split; FUNCTION secondStr(\$str) { split(" ",\$str)[2] } FUNCTION firstStr(\$str) { split(" ",\$str)[1] } FOR \$book IN document("book3.xml") //book[price div(100) .>. "8" AND secondStr(author/name) = "Melville" OR firstStr(author/name) = "Herman"] RETURN <book>\$book</book>

0	book.xml
1	/book
1.1	/book/price
1.2	/book/author
1.2.1	/book/author/full_name
1.2.1.1	/book/author/full_name/first_name
1.2.1.2	/book/author/full_name/last_name
1.3	/book/title
1.4	/book/year
1.5	/book/publisher
1.6	/book/editor
1.6.1	/book/editor/affiliation
1.6.2	/book/editor/full_name

그림 13 인덱스가 부여된 전역 문서 DTD 예

를 가지고 있지만, Book3.DTD에는 없다. 이 경우에 “price” 요소를 포함하고 있는 전역 질의어에 대해서 Book3를 위한 지역 질의어는 생성되지 않는다. 왜냐하면 Book3.DTD는 “price” 요소에 해당하는 인덱스를 가지고 있지 않기 때문이다. 그림 14와 같이 매핑된 정보를 이용하여 1:N과 N:1 매핑 경로를 가진 전역 질의어에 대해 생성된 지역 질의 생성 결과는 표 2와 같다.

5.2 석박사논문/논문지/연구보고서를 이용한 두개의 전역 DTD의 전역 질의에 대한 지역 질의 생성 및 실행 결과

5.2.1 기본적인 정보로 구성된 전역 DTD

표 3 Sample1.DTD : 전역 DTD 예

```

<?xml version="1.0" encoding="euc-kr"?>
<!ELEMENT 문서 (제어번호?, 제목, 요약서?, 본문,
참고문헌)>
<!ELEMENT 제어번호 (#PCDATA)>
<!ELEMENT 제목 (#PCDATA)>
<!ELEMENT 요약서 (발행일, 인물정보+, 요약문+,
키워드*)>
<!ELEMENT 날짜 (#PCDATA)>
<!ELEMENT 인물정보 (성명+, e-mail?)>
<!ELEMENT 성명 (#PCDATA)>
    
```

```

<!ELEMENT 인물정보 (성명+, e-mail?)>
<!ELEMENT 성명 (#PCDATA)>
<!ELEMENT e-mail (#PCDATA)>
<!ELEMENT 요약문 (#PCDATA)>
<!ELEMENT 키워드 (국문키워드|영문키워드)>
<!ELEMENT 국문키워드 (#PCDATA)>
<!ELEMENT 영문키워드 (#PCDATA)>
<!ELEMENT 본문 (부편+ | 장편+ |img)>
<!ELEMENT 부 (제목?, 장편*)>
<!ELEMENT 장 (제목?, 절편*)>
<!ELEMENT img (name | src)>
<!ELEMENT 번호 (#PCDATA)>
<!ELEMENT 절편 (#PCDATA)>
<!ELEMENT 참고문헌 (제어번호)+>
<!ATTLIST 키워드 언어 CDATA #IMPLIED>
<!ATTLIST img id CDATA #IMPLIED>
    
```

5.2.1.1 등록된 문서 중 발행일이 “1997. 12. 15”인 문서의 제목과 발행일을 검색하라.

전역 질의	FOR \$doc IN document("M2_small.xml") WHERE \$doc//발행일 = "1997. 12. 15" RETURN <발행>\$doc/제목, \$doc//발행일</발행>
생성된 지역 질의	FOR \$doc IN document("Jour_small.xml") WHERE \$doc//전방/논문지정보/출판일 = "1997. 12. 15" RETURN <발행>\$doc/전방/제목그룹/제목, \$doc//전방/논문지정보/출판일</발행>
	FOR \$doc IN document("Rep_small.xml") WHERE \$doc//요약서/일반정보/발행일 = "1997. 12. 15" RETURN <발행>\$doc/요약서/일반정보/중점과제명,\$doc//요약서/일반정보/발행일</발행>
	FOR \$doc IN document("Suk_small.xml") WHERE \$doc//전방/논문정보/날짜 = "1997. 12. 15" RETURN <발행>\$doc/전방/논문정보/제목그룹/제목, \$doc//전방/논문정보/날짜</발행>
질의결과 없음	
	<발행> <중점과제명 언어="kor"> SGML 문서를 기반으로 하는 정보 서비스에 관한 연구

[Book1.Index]

0	Book1.xml
1	/bib/book
1.1	./price
1.2	./author
1.2.1.1	./author/first
1.2.1.2	./author/last
1.3	./title
1.4	./@year
1.5	./publisher
1.6	./editor
1.6.1	./editor/affiliation
1.6.2	./editor/last
1.6.2	./editor/first

[Book2.Index]

0	Book2.xml
1	/arts/book
1.2	./author
1.2.1.1	./author/firstname
1.2.1.2	./author/lastname
1.3	./title
1.4	./@year
1.5	./publisher

[Book3.Index]

0	Book3.xml
1	/bookstore/book
1.1	./price div(100)
1.2	./author
1.2.1.1	./name fstring
1.2.1.2	./name lstring
1.3	./title

그림 14 세 지역 DTD에 대해 인덱스 된 DTD 트리

결 과	</중점과제명> <발행일> 1997. 12. 15 </발행일> </발행>
	질의결과 없음

5.2.1.2 등록된 문서 중 인물정보/성명이 "이규철"인 사람의 성명과 email을 검색하라.

전역 질의	FOR \$doc IN DISTINCT document ("M2_small.xml")//인물정보[성명="이규철"] RETURN <인물정보>\$doc/성명 , \$doc/email</인물정보>
생성된 지역 질의	FOR \$doc IN DISTINCT document("Jour_small.xml")//전방/저자그룹/저자[이름="이규철"] RETURN <인물정보>\$doc/이름 , \$doc/email</인물정보>
	FOR \$doc IN DISTINCT document("Rep_small.xml")//요약서/연구원정보/연구책임자[성명="이규철"] RETURN <인물정보>\$doc/성명 , \$doc/email</인물정보>
	생성된 질의 없음
결 과	질의결과 없음
	<인물정보> <성명 언어="kor"> 이규철 </성명> <email> nonno@pllab.com </email> </인물정보>
	질의결과 없음

5.2.1.3 등록된 문서 중 제목이 "SGML 문서를 기반으로 하는 정보 서비스에 관한 연구"이거나 "음향 분절모델과 상대 거리에 기반한 평활화를 이용한 한국어 고립 단어 인식에 관한 연구"인 문서를 검색하라.

전역 질의	FOR \$doc IN document("M2_small.xml") WHERE \$doc[제목 = "SGML 문서를 기반으로 하는 정보 서비스에 관한 연구" OR 제목 = "음향 분절모델과 상대 거리에 기반한 평활화를 이용한 한국어 고립 단어 인식에 관한 연구"] RETURN <문서>\$doc</문서>
생성된 지역 질의	FOR \$doc IN document("Jour_small.xml") WHERE \$doc[전방/제목그룹/제목 = "SGML 문서를 기반으로 하는 정보 서비스에 관한 연구" OR 전방/제목그룹/제목 = "음향 분절모델과 상대 거리에 기반한 평활화를 이용한 한국어 고립 단어 인식에 관한 연구"] RETURN <문서>\$doc</문서>
	FOR \$doc IN document("Rep_small.xml") WHERE \$doc[요약서/일반정보/중점과제명 = "SGML 문서를 기반으로 하는 정보 서비스에 관한 연구" OR 요약서/일반정보/중점과제명 = "음향 분절모델과 상대 거리에 기반한 평활화를 이용한 한국어 고립 단어 인식에 관한 연구"] RETURN <문서>\$doc</문서>

결 과	이용한 한국어 고립 단어 인식에 관한 연구"] RETURN <문서>\$doc</문서>
	FOR \$doc IN document("Suk_small.xml") WHERE \$doc[전방/논문정보/제목그룹/제목 = "SGML 문서를 기반으로 하는 정보 서비스에 관한 연구" OR 전방/논문정보/제목그룹/제목 = "음향 분절모델과 상대 거리에 기반한 평활화를 이용한 한국어 고립 단어 인식에 관한 연구"] RETURN <문서>\$doc</문서>
결 과	<문서> <제목> 음향 분절모델과 상대 거리에 기반한 평활화를 이용한 한국어 고립 단어 인식에 관한 연구 </제목> </문서>
	<문서> <중점과제명 언어="kor"> SGML 문서를 기반으로 하는 정보 서비스에 관한 연구 </중점과제명> </문서>
	질의 결과 없음

5.2.2 상세 정보를 포함하는 전역 DTD

표 4 Sample2.DTD : 전역 DTD 예

<pre> <?xml version="1.0" encoding="euc-kr"?> <!-- <!ENTITY % doctype "석박사논문"> <!ENTITY % doctype "논문"> <!ENTITY % doctype "연구보고서"--> <!-- Mixed Parameter Entity Declarations used in the DTD --> <!ENTITY % m.date "#PCDATA"> <!ENTITY % p.float "그림그룹 각주"> <!ENTITY % s.zz "단락 인용단락 참고문헌 %p.float;"> <!ENTITY % m.sec "(제목?, (%s.zz)*, 절*)"> <!ENTITY % aid "id ID #IMPLIED"> <!ELEMENT 문서 (제어번호?, 요약서?, 목차, 본문, 부록물?, 부속물?)> <!ELEMENT 제어번호 (#PCDATA)> <!-- 요약서 --> <!ELEMENT 요약서 (일반정보, 인물정보, 본문요약, 키워드)*> <!ELEMENT 일반정보 (과제코드?, 발행일)> <!ELEMENT 과제코드 (#PCDATA)> <!ELEMENT 발행일 (%m.date:)> <!ELEMENT 인물정보 (성명+, 주민번호?, email?, 전화번호?)> <!ELEMENT 주민번호 (#PCDATA)> <!ELEMENT email (#PCDATA)> <!ELEMENT 전화번호 (#PCDATA)> <!ELEMENT 성명 (#PCDATA)> <!ELEMENT 본문요약 (%m.sec:)+> <!ELEMENT 요약문 (#PCDATA)> <!ELEMENT 키워드 (국문키워드 영문키워드)+> </pre>
--

```

<!ELEMENT 국문키워드 (#PCDATA)>
<!ELEMENT 영문키워드 (#PCDATA)>
<!-- 목차 ..... -->
<!ELEMENT 목차 (제목?, 목차내용)+>
<!ELEMENT 제목 (#PCDATA)>
<!ELEMENT 목차내용 EMPTY>
<!-- 본문 ..... -->
<!ELEMENT 본문 (부_+ | 장_+ | %m.sec; | img)>
<!ELEMENT 부_ (번호)>
<!--ELEMENT 부_ (번호, 제목, 장)-->
<!ELEMENT 장_ (번호, %m.sec;)>
<!ELEMENT 번호 (#PCDATA)>
<!ELEMENT img EMPTY>
<!-- 부록물 엘리먼트 ..... -->
<!ELEMENT 부록물 (%m.sec;)*>
<!-- 부속물 엘리먼트 ..... -->
<!ELEMENT 부속물 (참고문헌)*>
<!ELEMENT 참고문헌 (제목)>
<!-- ..... -->
<!-- Attribute List Declarations ..... -->
<!-- ..... -->
<!ATTLIST 부록물 %a.id;>
<!ATTLIST img %a.id; name ENTITY #IMPLIED src
CDATA #IMPLIED>
<!ATTLIST 인물정보 유형 CDATA #IMPLIED>
<!ATTLIST 키워드 언어 CDATA #IMPLIED>
    
```

5.2.2.1 문서 중 인물정보/성명을 검색하되 **인물정보 /@유형별로** 묶어 출력하라.

전역 질의	<pre> FOR \$i IN DISTINCT document("M2_ middle.xml")// 인물정보/@유형 LET \$b := document("M2_middle.xml")// 인물정보[@유형 = \$i] RETURN <유형별> \$i, \$b/성명/text() </유형별> </pre>
지역 질의	<pre> FOR \$i IN DISTINCT document("Jour_middle. xml") //전방/저자그룹/저자/역할 LET \$b := document ("Jour_middle.xml")// 전방/저자그룹/저자[역할 = \$i] RETURN <유형별>\$i,\$b/이름/text()</유형별> FOR \$i IN DISTINCT document("Rep_middle. xml")//연구원정보/@유형 LET \$b := document ("Rep_middle.xml")// 연구원정보[@유형 = \$i] RETURN <유형별>\$i,\$b/연구책임자/ 성명</유형별> </pre>
결과	<pre> <유형별> <역할> 주연구원 </역할> 설아침 감기훈 </유형별> <유형별> <역할> 참여연구 </역할> 최성희 이정숙 </유형별> <!-- 질의결과 없음 --> <!-- 질의결과 없음 --> </pre>

6. 결론 및 향후 연구

본 논문에서는 분산 문서 환경에서의 의미, 구조적인 이질성 문제를 해결하기 위해 분산 데이터의 의미론적인 정보도 함께 포함하는 메타 인터페이스인 DDXMI를 제안하고 이를 이용한 통합 검색 시스템을 구현 하였다. DDXMI는 DTD 트리를 이용하여 각 요소의 의미를 고려한 인덱스 번호를 부여하는 방법으로 간단하게 생성 되고, 이에 따라 전역 질의어는 DDXMI를 이용하여 각 지역 문서를 위한 지역 질의어로 자동으로 변환된다. 이렇게 변환된 각 지역 문서를 위한 질의어들은 각자의 문서 엔진에서 처리되고 결과를 생성한다.

본 시스템은 자바 서블릿 서버와 JavaCC 컴파일러를 이용한 NT 환경에서 구현되었다. XML로 표현 가능한 어떠한 문서라도 변경이나 재구성 없이 본 시스템에 사용될 수 있다. DDXMI를 이해하고 생성하는 작업은 XML에 대한 사전 지식을 가지고 있는 사용자라면 본 시스템을 사용하는데 어려움이 없을 것으로 판단된다. 전역 DTD를 사용자가 편리한대로 생성하여 사용하기 편리한 장점이 있다.

향후 연구과제로는 서로 다른 구조를 가진 XML 스키마에 지역 질의를 생성할 수 있는 방법과, DTD 트리 내에 너무 많은 노드들이 존재할 때 하나의 노드 단위의 조작 대신에 서브트리 단위의 조작이 가능하도록 하는 연구가 필요하다.

참고 문헌

- [1] C. Parent and S. Spaccapietra, "Issues and Approaches of Database Integration. Communications of the ACM," 41(5):166-178, 1998.
- [2] Amit P. Sheth, "Changing focus on interoperability in information systems : from system, syntax, structure to semantics," In M F Goodchild, M J Egenhofer, R Fegeas and C A Kottman (eds), Interoperating Geographic Information Systems. Kluwer. 1998.
- [3] Online Computer Library Center, "Dublin Core Metadata Element Set : Reference Description," 1997, Office of Research and Special Projects, Dublin, Ohio. [http:// www.oclc.org:5046/research/dublin_core/](http://www.oclc.org:5046/research/dublin_core/)
- [4] Beard K., Smith T, "A framework for meta-information in digital libraries," in Sheth A, Klas W (eds) Multimedia Data Management : Using Metadata to Integrate and Apply Digital Media. McGraw Hill: 341-365. 1998.
- [5] Gnther O, Voisard A, "Metadata in geographic and environmental data management," in Sheth A, Klas W (eds) Multimedia Data Management: Using Metadata to Integrate and Apply Digital

Media. McGraw Hill: 57-87. 1998.

[6] O.J. Reichman, et al, "A Knowledge Network for Biocomplexity : Building and Evaluating a Meta-data-based Framework for Integrating Heterogeneous Scientific Data", <http://www.nceas.ucsb.edu/>

[7] XML Metadata Interchange (XMI), <http://omg.org/technology/documents/formal/xmi.html>

[8] XQL (XML Query Language), Aug. 1999, <http://www.ibiblio.org/xql/xql-proposal.html>

[9] S. Ram and V. Ramesh, "Schema Integration : Past, Current and Future," in A. Elmagarmid, M. Rusinkiewica, and A. Sheith, editors, Management of Heterogeneous and Autonomous Database Systems, page 119-155. Morgan Kaufmann Publishers, 1999.

[10] L.M.Haas, R.J.Miller, B.Niswonger, M.Tork Roth,P. M.Schwarz and E.L.Wimmers, "Transforming Heterogeneous Data with Database Middleware : Beyond Integration," Bulletin of IEEE Computer Society Technical Committee on Data Engineering, 1999.

[11] R.J.Miller, L.M.Haas and M.A.Hernandez, "Schema Mapping as Query Discovery," in Proceedings of the 26th VLDB Conference. Cairo Egypt, 2000.

[12] R.J.Miller, "Using Schematically Heterogeneous Structures," SIGMOD '98 Seattle WA, USA. ACM 0-89791-995-5. 1998.

[13] S.Cluet, C.Delobel; J.Simeon and K.Smaga, "Your Mediators Need Data Conversion," in ACM SIGMOD Conference, pages 177-188, 1998.

[14] S.Abiteboul, S.Cluet and T.Milo, "Correspondence and Translation for Heterogeneous Data," in Proc. Of the Intl Conf. on Database Theory (ICDT), pages 351-363, 1997.

[15] T.Milo and S.Zohar, "Using Schema Matching to Simplify Heterogeneous Data Translation," in Proc. of the Intl Conf. on VLDB, pages 122-133, NY, 1998.

[16] Y. Papakonstantinou, H. Garcia-Molina and J.Ullman, MedMaker, "A Mediation System Based on Declarative Specifications".

[17] C. Baru, A. Gupta, B. Ludscher, R. Marciano, Y. Papakonstantinou, P. Velikhov, V. Chu, "XML-Based Information Mediation with MIX. Exhibition program," in ACM Conf. on Management of Data, SIGMOD '99, Philadelphia, 1999.

[18] The Extensible Stylesheet Language (XSL), <http://www.w3.org/Style/XSL/>

[19] Sheth A, "Data semantics : What, where, and how?," in Meersman R, Mark L (eds) Database Application Semantics, Chapman and Hall : 601-610. 1997.

[20] Brickley D, R.V Guha, eds, "Resource Description Framework (RDF) Schema Specification," W3C Proposed Recommendation, March 1999, <http://www.w3.org/1999/TR/PR-rdf-schema>.

[21] Knowledge Sharing Effort, <http://www-ksl.stanford.edu/knowledge-sharing>.

[22] Intelligent Integration of Information, <http://mole.dc.isx.com/I3>.

[23] Digital Librabry Initiative, http://www.cise.nsf.gov/iis/dli_home.html

[24] A. Gupta, B. Ludscher, M. E. Martone, "Knowledge-Based Integration of Neuroscience Data Sources," 12th Intl. Conference on Scientific and Statistical Database Management (SSDBM), Berlin, Germany, IEEE Computer Society, July, 2000.

[25] B. Ludscher, A. Gupta, M. E. Martone, "Model-Based Mediation with Domain Maps," 17th Intl. Conference on Data Engineering (ICDE), Heidelberg, Germany, IEEE Computer Society, April 2001.

[26] B. Ludascher, R. Himmeroder, G. Lausen, W. May, and C. Schlepphorst, "Managing Semistructured Data with FLORID : A Deductive Object-Oriented Perspective," Information Systems, 23(8):589-613, 1998.

[27] A. Farquhar, R.Fikes and J.Rice, "The Ontliqua Server : A Tool for Collaborative Ontology Construction," International Journal of Human-Computer Studies, 46 : 707-728, 1997.

[28] V. K. Chaudhri, A. Farquhar, et al. "OKBC : Open Knowledge Base Connectivity 2.0," Technical report KSL-98-06, Knowledge System Laboratory, Stanford, July 1997.

[29] 이명철 외, DataBlender : "XML 기반 가상 데이터베이스 통합 시스템", 데이터베이스연구, 제19권 1호, 2003. 3. pp 15-27.

[30] 이경하 외, "XMF: XML 기반 분산 이질 정보 자원의 통합 프레임워크", KDBC2000, pp. 262-270, 2000.

[31] 정도현, "시맨틱웹을 위한 온톨로지 언어와 구현사례 연구", 정보관리연구, vol 34, no. 3, 2003, pp. 87-109



류 성 준
 2001년 2월 연세대학교 전산학과 졸업.
 2003년 2월 연세대학교 대학원 전산학과 졸업(석사). 2003년 3월~현재 포스테이타 e-biz 연구소 근무. 관심분야는 XML, 정보검색, 소프트웨어공학



황 재 문
 2001년 2월 연세대학교 수학과 졸업. 2003년 8월 연세대학교 전산학과 석사 졸업. 2004년 1월~현재 LG 전자 정보통신사업본부 근무. 관심분야는 XML, 정보검색, 임베디드 소프트웨어



김 태 훈

2002년 9월 연세대학교 전산학과 졸업.
 현재 연세대학교 전산학과 대학원 석사
 4학기. 관심분야는 프로그래밍언어, XML,
 정보검색



남 영 광

1978년 2월 연세대학교 수학과 졸업. 1985
 년 2월 한국과학기술원 전산학과 졸업
 1992년 12월 Northwestern University
 전산학과 졸업. 1993년 1월~1994년 12
 월 시스템공학연구소 선임연구원. 1995년
 3월~현재 연세대학교 전산학과 교수 관
 심분야는 프로그래밍언어, 소프트웨어공학, 정보검색, XML