

의미적 제약조건을 보존하는 DIDL 매핑 시스템의 설계 및 구현

정회원 송 정 석*, 김 우 생**

A design and implementation of DIDL mapping system preserving semantic constraints

Jung-suk Song*, Woo-saeng Kim** *Regular Members*

요 약

최근 XML 문서가 인터넷 기반의 애플리케이션 간의 자료 저장 및 교환을 위한 표준으로서 부상함에 따라 XML 문서의 저장 및 관리에 대한 연구가 활발히 이루어지고 있다. 하지만 대부분의 연구들은 인스턴스 및 DTD 기반의 매핑 기술에 관한 것으로, 문서의 구조적 변환에만 초점이 맞추어져 있었다. 현재는 XML 스키마 기반의 XML 문서 이용이 보편화되고 있는 추세이며 구조적 변환 이외에도 의미의 보존이 아울러 요구되고 있다. 이에 본 논문에서는 XML 스키마에 기반한 MPEG-21의 DIDL(Digital Item Declaration Language) 언어를 응용 도메인으로 설정하고, 의미적 제약조건을 확장하여 보존할 수 있는 매핑 모델을 제안하였다. 기존의연구결과를 제시된 도메인에 적절하게 확장하여 전처리 단계에서 적용한 후, 의미 정보의 보존을 위한 다양한 후처리 과정을 설계단계에 적용하였다. 구현에 적용된 시스템의 구성 및 제안 알고리즘을 소개하고, 구현 환경 및 상세한 의미적 확장 기법을 제시하였으며, 테이블과 질의 예제를 통한 질의처리 예제를 보였다.

Key Words : Digital item, XML Schema, MPEG-21, Semantic constraints, Relational database

ABSTRACT

Recently, XML has been emerging as a standard for storing and exchanging of data for various distributed applications based on the Internet. Since there are increasing demands to store and manage XML documents, a lot of research works are going on this area to develop new tools and techniques based on the XML. However, most of the researches are concentrated on mapping techniques based on instance or DTD, and the main focus is on structural transformation. Current trend of research is toward the usage of XML documents based on XML schema, and demands not only conversion of structure but also preservation of the semantic constraints. This paper sets up the using of DIDL standing on the basis of XML schema from MPEG-21 as an application domain, and proposes the mapping model that can preserve semantic constraints in addition. We expand previous research techniques in the preprocessing step for the specific domain, and then, apply various new mapping methods in the postprocessing step. We present and discuss the system architecture for implementation, and introduce the algorithms and present implementation environment and semantic extension methodology in detail. Finally we show actual table and query processing based on our proposal.

I. 서 론

최근 XML 문서가 문서간의 데이터 교환을 자동

화하고, 안전한 자료의 교환을 위한 도구로서 평가 받음에 따라, 인터넷 기반의 상업 애플리케이션을 위한 표준 데이터 교환 형식을 지원하는 표준으로

* 한화S&C(주) 기술연구소 (songjs@hanwha.co.kr)

** 광운대학교 컴퓨터공학부 멀티미디어연구실 (kwsrain@cs.kw.ac.kr)

논문번호 : #030544-1208, 접수일자 : 2003년 12월 11일

*본 연구는 산업자원부 중기거점 연구과제 (10006936) 및 2004학년도 광운대 교내 학술 연구비 지원에 의해 수행되었습니다.

서 부상하고 있다[1]. 이에 따라 비즈니스 응용프로그램에서 XML을 이용하여 자료를 표현하고 저장하려는 추세가 두드러지고 있으며, 애플리케이션에 저장되어 있던 방대한 양의 데이터와 XML 문서 사이의 호환 문제가 급격히 부상하게 되었다. 아울러 XML과 관련된 데이터 관리의 필요성이 제기됨으로써 대량의 XML 문서들을 효율적으로 저장하고 관리하는 시스템에 대한 요구가 증가하고 있다.

기존의 관계형 데이터베이스에 XML 문서를 저장하는 경우, 서로 상이한 모델에 기반한 구조를 이용하기 때문에 상호 모델간의 매핑하는 방법이 매우 복잡하다. 이러한 문제점을 해결하기 위한 연구가 다양한 방식으로 진행되어 왔는데, Instance 기반의 변환방법[2,3]과 DTD 기반의 변환 방법[4,5]이 주류를 이루면서 구조적인 변환을 위한 다양한 해법을 제시하고 있다.

또한 풍부한 데이터의 표현이 가능하고 다양한 타입이 지원되는 XML 스키마[6]의 이용이 각종 응용분야에 적용되고 있다. XML 스키마에서는 제공되는 데이터타입이 풍부하기 때문에 보다 많은 고려사항이 뒷받침되어야 한다. [7]에서는 XML 스키마 기반의 문서를 변환하여 관계형 데이터베이스에 저장하기 위한 연구결과를 제시했으며, [8]에서는 최적의 매핑을 선택하는 비용기반의 연구가 이루어졌다.

[5]에서는 XML 문서의 구조적인 내용만을 변환하는 방식에서 벗어나 의미적인 제약조건을 포함하는 변환방식을 소개하였다.

멀티미디어 프레임워크의 제정을 위한 표준인 MPEG-21[9]에서는 기본 단위를 디지털 아이টে姆으로 정의하고 이를 XML 스키마에 기반한 고유의 언어인 DIDL[10]을 통하여 이용한다. 하지만 DIDL문서는 선언방법에 대한 표준만을 정의하고 있기 때문에, 선언된 데이터의 저장이나 검색 등에 대한 고려를 위하여 [11]에서는 DIDL 문서를 관계형 데이터베이스에 저장하기 위한 스키마를 설계하고 기존의 데이터와 연계하여 검색 및 처리할 수 있는 모델을 제시하였다.

본 논문에서는 XML 스키마를 토대로 표준으로 정의된 DIDL 문서를 응용 도메인으로 설정하여 관계형 스키마로 매핑하는 방법론을 제안하고자 한다. 이때 DIDL 스키마에서 정의된 다양한 의미적 정보를 보존하기 위한 확장 모델을 설계하고, 이를 바탕으로 시스템을 구현하였다.

본 논문의 구성은 다음과 같다. 제2절에서는 관

련연구로서 XML 문서의 관계형 데이터베이스로의 매핑에 대한 기존의 연구에 대하여 설명한다. 제3절에서는 DIDL 문서를 처리하기 위한 모델의 설계과정으로서 전처리 과정은 기존의 연구를 최대한 이용하고 의미적 정보의 보존을 위한 확장된 고려사항을 제시한다. 제4절에서는 확장된 설계를 바탕으로 구현된 내용을 설명한다. 마지막으로 제5절에서는 결론을 맺는다.

II. 관련연구

관계형 데이터베이스를 이용하여 XML 문서를 생성하고 질의하기 위한 많은 연구가 제안되었다. Instance 기반의 저장기법에 대하여 살펴보면 [2]에서는 스키마 정보가 없는 경우에 XML 문서로부터 스키마 정보를 추출해 내어 관계형 테이블을 생성해내는 방법을 이용한다. [3]에서 제안한 Edge 방식은 스키마 정보는 무시하고, XML 문서를 구성하는 엘리먼트들, 즉 노드와 간선(Edge)들만을 저장하는 방법이다. Binary 방식은 간선의 이름별로 별도의 이진 테이블을 만드는 방법이며, 이렇게 만들어진 모든 테이블들을 Outer Join으로 결합 시키는 것이 Universal 방식이다.

DTD 기반의 저장기법에 대한 대표적 사례로서 [4]에서는 DTD에 표현된 정보를 이용하여 Element Graph를 만들고, Element Graph를 통하여 모든 엘리먼트의 테이블을 생성하는 Basic 인라인 방식을 제시하고 있다. Shared 방식은 다른 테이블과 공유될 수 있는 엘리먼트를 만나게 되면 인라인 시키지 않고 독립된 테이블로 처리하고, Set-valued 속성과 재귀상황이 아닌 경우 다른 테이블에 인라인 시키는 Hybrid 방식을 추가적으로 제시하고 있다.

상용 DBMS에서의 XML 지원기능을 살펴보면, Oracle은 XSQL, SQL Server는 XDR(Xml Data Reduced), DB2는 DAD(Document Access Definition) 등의 매핑 구조를 제공하며, 사용자로 하여금 관계형 데이터의 XML 뷰를 생성할 수 있도록 제공하고 있다. 하지만 뷰에 대한 질의 지원이 미비하고, 중첩된 구조의 원활한 지원이 제공되지 않는 등의 많은 제약조건을 안고 있다.

[5]는 DTD에서 각종 제약조건에 해당하는 조건들을 추출하여 관계형 스키마로 변환할 때, 제약조건을 반영하여 테이블을 구성하는 연구이다. 이용되는 제약조건으로는 기수 제약조건, 포함 종속, 단일 생성 종속, 투플생성 종속 등이 있다. 기존의 연구

가 구조적인 변환에 중점을 두고 있는 반면, 의미적 조건을 고려하여 변환할 수 있도록 하였다.

대부분의 연구들이 DTD를 기반으로 변환하는 연구이기 때문에 다양한 타입을 지원하는 XML 스키마 기반의 경우에는 적용에 어려움이 있다. [7]의 연구에서 XML 스키마 기반의 변환을 위한 연구가 있었으며, 특히 [8]에서는 주어진 애플리케이션을 위한 최적의 매핑을 선택하는 비용 기반의 XML 저장 매핑 엔진을 제안하였다. XML 스키마, 자료통계, 질의부담을 바탕으로 목표 애플리케이션을 모델링하고, 재구성한 후, 유도된 구성 중에서 관계형 최적기를 통한 최상의 것을 선택하는 방법을 이용한다.

이상의 연구들은 XML 문서의 구조적인 내용만을 변환에 반영하지만 [5]에서는 CPI 알고리즘 등을 제안하여 의미적인 제약조건을 포함하는 변환방식을 소개하여 보다 많은 의미적 정보를 손실 없이 저장할 수 있도록 하였다.

[11]에서는 MPEG-21의 DIDL 문서를 관계형 데이터베이스에 저장하여 사용할 수 있는, 특정 응용도메인에서의 최적화된 변환모델을 제안하였다. DIDL 문서에 표현된 내용을 저장하기 위하여 관계형 스키마로 매핑하였으며, 각각의 엘리먼트를 위한 테이블에 상호연계를 위한 별도의 필드를 할당하여 노드 간의 관계표현을 보존하였고, SQL 문을 이용하여 데이터의 핸들링 및 기존 자료와의 연계를 도모할 수 있도록 하였다.

III. 관계형 스키마 설계

1. DIDL 문서

디지털 아이템은 MPEG-21에서 처리되는 개체의

Digital Item = resources + metadata+ structure

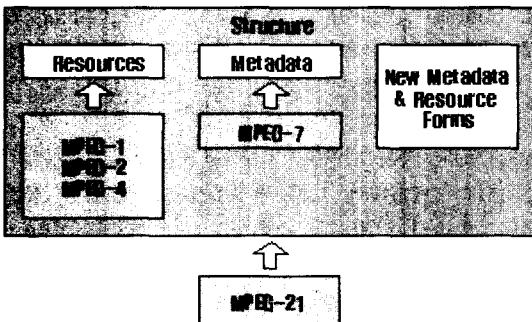


그림 1. MPEG-21의 디지털 아이템 구성방법

기본적 단위로서 멀티미디어 콘텐츠와 같은 개념으로 사용된다. 이는 일종의 구조적인 멀티미디어 객체로서, 멀티미디어 리소스 자체를 나타내는 물론, 메타데이터 및 구조 또한 포함하는 것으로서 그림 1과 같은 형태로 표현한다.

MPEG-21에서는 디지털 아이템의 선언을 위하여 EBNF(Extended Backus-Naur Form) 표기법을 통하여 모델을 설명하고, XML 스키마를 이용하여 디지털 아이템을 구성하는 각 엘리먼트의 상세한 의미를 표현한다. DIDL 스키마에 정의되어 있는 엘리먼트들을 위한 XML 스키마 표현은 [10]에 자세히 설명되어 있다.

2. DIDL 문서의 테이블 매핑

2.1. 엘리먼트의 매핑

DIDL 문서는 DIDL 스키마를 통하여 매우 체계적으로 구성되어 있고, 서브엘리먼트 대부분이 다른 엘리먼트를 참조하는 구조이기 때문에 상대적으로 쉽게 매핑할 수 있다. 본 논문에서는 [4]에서 제안한 인라이닝 기법을 수용하여 엘리먼트의 In-degree가 1이고, 서브엘리먼트의 개수가 각각 1개인 경우만 상위 노드에 인라인 시키고, 나머지 경우에는 별도의 테이블로 매핑한다.

엘리먼트 그래프의 구조관계 중에서 Descriptor와 Component 사이의 순환관계와, Item, Container, Descriptor 등의 재귀관계 역시 별도의 테이블로 매핑한 후 참조하는 방식을 사용한다.

2.2. 애트리뷰트의 매핑

XML 스키마에서는 같은 이름을 가진 속성이 다수가 발생하지 않기 때문에 기존연구와 마찬가지로 DIDL내의 모든 엘리먼트에 대한 애트리뷰트는 필드로 매핑한다.

AttributeGroup의 경우에는 그룹 내의 애트리뷰트들을 참조하는 엘리먼트의 개별적인 필드로 일일이 생성하여 매핑한다. "ID_ATTRS"경우가 해당된다.

2.3. 값의 매핑

Statement 엘리먼트는 타입정보와 실제 데이터를 저장할 수 있는 엘리먼트로서, 텍스트, XML 등의 정보가 삽입될 수 있는 특수한 엘리먼트이다. BLOB 필드형태를 통하여 대용량 데이터 공간을 확보할 수 있도록 특별하게 처리하였다.

```
CREATE TABLE STATEMENT (
    value [blob]
    mimeType [varchar] (255)
)
```

2.4. 관계 테이블

DIDL 문서의 각 엘리먼트 중에서 별도의 테이블로 매핑된 엘리먼트들 간의 상하관계를 표현하기 위하여 본 논문에서는 [3]에서 제안한 Edge 방식을 확장하여 별도의 관계테이블을 생성한 후, 엘리먼트 간의 관계정보를 저장한다. 관계테이블의 형태는 그림 2와 같다.

--	--	--	--	--	--

그림 2. 관계 테이블

ParentType은 부모 노드가 어떤 엘리먼트인지 표시하게 되고, ParentID는 부모 노드의 고유값을 나타낸다. ChildType과 ChildID 또한 자식노드의 형태와 고유값을 나타낸다. Order는 엘리먼트 내에서 서브엘리먼트간의 순서를 표현하는 정보로서 <sequence> 태그의 의미정보를 유지하기 위한 용도로서 이용된다.

본 논문에서는 Edge 방식의 Flag 정보를 확장하여, 노드의 상하방향 탐색을 모두 지원하기 위하여 상하 노드의 관계를 보존하였다. 이는 DIDL 문서가 Item 기반으로 이용되며, DIA (Digital Item Adaptation) 등의 적용엔진을 통한 메타데이터 변환시 Item 기반의 처리가 빈번하기 때문에 이를 최적화하기 위하여 Item 기반의 양방향 처리가 가능하도록 하기 위한 방안이다.

또한 부모 엘리먼트의 ID 정보뿐만 아니라 타입 정보를 별도로 저장하고 있기 때문에, Descriptor 테이블의 경우처럼 부모노드로서 나타날 수 있는 엘리먼트가 9개나 되는 경우, 부모 엘리먼트의 형태를 이는 경우, 해당 형태의 테이블과 Descriptor 테이블 만들을 조인하면 되기 때문에 추가적인 최적화 효과를 얻을 수 있다.

3. 제약조건 및 의미정보 부여

3.1. 기수 제약조건

XML 기반의 문서의 형태를 정의하는 부분에서

는 엘리먼트의 등장개수를 정의할 수 있다. DTD 선언에서는 그림 3과 같이 (0,1), (1,1), (0, 8), (1, 8) 등의 4가지 Cardinality 정의만이 가능하다.

(0,1)	?	0개 or 1개
(1,1)	Blank	1개
(0,∞)	*	0개 이상
(1,∞)	+	1개 이상

그림 3. DTD 기반의 Cardinality 조합

DIDL 엘리먼트 각각의 서브엘리먼트의 속성을 XML 스키마 형태로 표현하는 경우에는 최소 개수와 최대 개수를 보다 명시적으로 정의할 수 있다. XML 스키마에서 제공하는 minOccurs 속성과 maxOccurs 속성을 이용하여 표시하며, 관계형 스키마로 적용할 때에는, DTD에서 단순히 무한대만을 표시하는 것과는 달리, XML 스키마에서는 maxOccurs 속성에 'unbounded'를 통한 무제한 개수를 표현하는 방법과 명시적인 수치를 부여하는 방법으로 분리하여 생각해야 한다.

XML 스키마에서 엘리먼트에 대한 서브엘리먼트의 최소개수 혹은 최대개수를 명시적으로 정했을 경우에는, 최소개수 이하의 튜플 혹은 최대개수 이상의 튜플이 테이블에 저장되어서는 안 된다. 이를 위하여 본 논문에서는 DTD의 등장개수 조합을 확장하여 그림 4에 표현된 경우의 수를 모두 고려하는 방식을 적용하였다. XML 스키마에 표현된 Cardinality에 따라 의미를 파악하여 유효하지 않은 표현의 경우에 오류처리를 수행하고, 체크할 최소 최대항목을 결정한다.

(0,0)	Useless	없음
(0,N)	최대정의	최대개수
(0,∞)	관계형 스키마의 Default	없음
(N,0)	Non sense (min > max)	없음
(N,N)	최소정의, 최대정의	최소개수, 최대개수
(N,∞)	최소정의	최소개수
(∞,0)	Non sense (min > max)	없음
(∞,N)	Non sense (min > max)	없음
(∞,∞)	Ignore	없음

그림 4. XML 스키마 기반의 Cardinality 조합

기본적으로 관계형 데이터베이스에서는 튜플의 개수에 대한 선언은 별도로 하지 않으며 (0, 8)를 의미하게 된다. 이는 XML 스키마에서의 (minOccurs = 0, maxOccurs = unbounded)의 경우와 마찬가지로 볼 수 있다.

하지만 XML 스키마에서는 서브엘리먼트의 개수를 별도로 지정할 수 있기 때문에 (0, 8) 이외의 상황에 대한 추가적인 처리를 해주어야 XML 스키마에 표현된 튜플 개수에 대한 의미를 잃지 않고 보존할 수 있는 것이다.

이를 위하여 본 논문에서는 관계형 모델에서 제공하는 트리거 기능을 이용하여 테이블이 지정된 튜플 개수를 유지할 수 있도록 한다. 그림 5의 예제는 서브엘리먼트의 최대개수가 maxNumber로 설정된 경우, Insert, Update, Delete 처리가 이루어질 때 튜플의 개수가 최대값을 초과하면 Rollback 처리를 수행하고, 선언된 개수를 준수하면 Commit 처리를 수행하기 위한 문장을 표현하고 있다.

```
CREATE TRIGGER CheckMaxNumber
ON CheckTable
FOR INSERT, UPDATE, DELETE
AS
IF
(SELECT COUNT(*) FROM CheckTable
WHERE
(CheckTable.Parent ID = parent_id) > @@maxNumber)
Call Rollback_Process()
ELSE
Call Commit_Process()
```

그림 5. maxOccurs 속성을 고려한 트리거 문장

3.2. 키 제약조건

엘리먼트의 속성 중에서 ID 타입의 속성을 포함하는 경우에는 해당 애트리뷰트의 이름에 해당하는 테이블의 필드에 기본키를 할당하고 인덱스를 생성한다. 해당하는 엘리먼트는 Anchor, Annotation, Assertion, Choice, Component, Condition, Container, Declarations, Descriptor, DIDL, Item 등이며 다음과 같은 형태로 표현된다.

```
CREATE TABLE ANNOTATION (
[ID] [int] IDENTITY (0, 1) NOT NULL,
...
) ON [PRIMARY]
```

ID 타입의 속성을 포함하지 않는 Assertion, Choice, Condition, Declarations, DIDL, Reference, Resource, Selection, Statement 등은 추가로 ID 속성을 가진 필드를 추가하고 기본키로 설정한 후, 인

덱스를 생성한다.

3.3. 도메인 무결성 보장

속성, 값 등의 실제 데이터 값에 도메인이 설정되어 있는 경우, 해당 값의 범위를 제한하기 위하여 테이블을 생성할 때, 범위를 지정하여 제한된 범위 내의 값만이 테이블에 저장되도록 해주어야 한다. 이를 위하여 SQL문의 CHECK을 이용하여 해당 필드의 범위를 지정하도록 한다. 예를 들어 RT1 테이블의 order_id 필드의 값이 0보다 크고, 100보다 작다면 다음과 같이 표현될 수 있다.

```
ALTER TABLE RT1 ADD
CONSTRAINT [CK_RT1]
CHECK (order_id > 0 AND order_id < 100)
```

3.4. Nullable 처리

XML 스키마 기반 XML 문서의 경우, 엘리먼트와 애트리뷰트 모두가 테이블 내의 필드로서 매핑될 수 있기 때문에 Not null 특성을 고려해야 한다. 이를 위한 엘리먼트와 애트리뷰트의 속성에 따른 분류는 다음과 같다.

Attribute	use=required	→	TRUE
Attribute	use=optional	→	FALSE
Element	minOccurs ≥ 1	→	TRUE
Element	minOccurs=0	→	FALSE

애트리뷰트 내의 속성 중에서 use는 NULL의 의미에 해당하는 개념으로서, SQL 문으로 쉽게 적용할 수 있다. 엘리먼트의 경우에도 minOccurs 속성에 따라 동일한 방식을 적용한다. Assertion 엘리먼트의 매핑을 위한 문장은 다음과 같다.

```
CREATE TABLE ASSERTION (
[target] [varchar] (255) NOT NULL,
[true] [varchar] (255) NULL,
[false] [varchar] (255) NULL
)
```

IV. 시스템 구현

본 절에서는 정의된 모델에 따른 시스템을 구현한 과정을 설명한다.

1. 시스템 구성도

DIDL 문서는 최종적으로 그림 6과 같은 구조에

기반하여 설계 및 구현하였다. 최초 DIDL 문서를 저장하기 위한 저장기와 DI 기반의 디바이스에서 사용하기 위한 추출기로 크게 구성된다. DIDL Keeper는 DIDL 문서를 읽어 파싱과 유효성 검사를 거친 후, 저장 알고리즘에 따라서 관계형 데이터베이스에 변환하며, DIDL Extractor는 단순히 DIDL 문서를 복원하기 위한 모듈과 XPath 질의를 변환하여 처리하기 위한 모듈로 분리되어 처리된다.

- NotNull : 필드에 데이터를 저장할 때, 널의 허용여부 표시
- PK : Primary key로 이용되는 필드의 여부
- Unique : 중복 데이터의 허용여부 표시
- Min : 튜플의 최소 허용 개수
- Max : 튜플의 최대 허용 개수
- Flag : 저장된 데이터의 표현 형태 (integer, string, ...)
- Seq : sequence 태그의 복원을 위한 정보로서 서브엘리먼트간의 순서적 정보 기술
- Choice : choice 태그의 표현을 위한 정보로서 서브엘리먼트간의 선택적 정보 기술
- Default : 초기값이 설정되어 있을 경우 반영

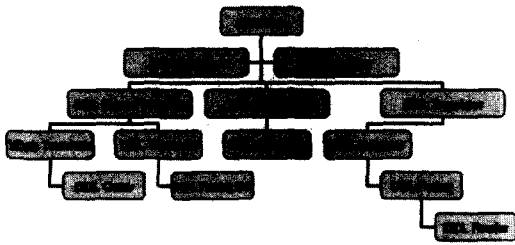


그림 6. 시스템 구성도

2. 구현환경

운영체제로는 시스템의 운영모듈이 존재하는 부분은 Windows XP를 사용하였고, 데이터베이스는 Windows 2003 Server에서 운영하였다. 전체적인 플랫폼은 Microsoft Visual Studio .NET 기반에서 이루어졌으며, 사용언어는 C#.NET을 이용하여 WinForm 기반 하에 구현하였다. 데이터베이스는 MS-SQL 2000을 사용하였고, 데이터베이스와의 연계를 위하여 ADO.NET을 이용하여 구현하였다. DIDL 문서의 파싱을 위하여 XML 형태의 DIDL 문서를 메모리에 적재하는 과정을 거치게 되는데, 이를 위하여 XMLReader 클래스를 사용하여 DataSet에 보관하여 처리하였으며, 메모리내의 DIDL 문서 핸들링을 위하여 DOM(Document Object Model)을 이용하여 알고리즘을 적용하였다.

3. 시스템 테이블

시스템의 운용을 위한 별도의 테이블을 구성하여 DIDL 고유의 특성을 반영한 처리를 수행하게 된다. 엘리먼트의 서브엘리먼트와 애트리뷰트를 처리하기 위한 추가적인 정보를 시스템 테이블로서 저장하기 위한 스키마의 형태와 필드에 대한 설명은 다음과 같다.

sys_subelements(Table, Field, Type, NotNull, PK, Unique, Min, Max, Flag, Seq, Choice, Default)
■Table : 엘리먼트를 매핑한 테이블의 이름
■Field : 서브엘리먼트와 애트리뷰트 및 값의 이름
■Type : 엘리먼트, 애트리뷰트, 값의 분류 목적 형태

<sequence> 태그를 위한 'Seq' 필드는 XML 문서에서 표현된 순서적 의미를 관계형 스키마에 적용하기 위한 추가적인 정보로서, <sequence> 태그에 표현되어 있는 엘리먼트 내의 서브엘리먼트 및 애트리뷰트의 순서에 따라서 추후 DIDL 문서로 복원시에 재구성을 함으로써 순서정보를 보존할 수 있게 된다.

<choice> 태그를 위한 'Choice' 필드는 선택적 의미를 보존하기 위한 것으로서 데이터베이스에 연산을 적용할 때 특별한 조건을 추가적으로 고려해야 한다. Descriptor 테이블의 Reference, Component, Statement 서브엘리먼트들의 경우, 세 가지 서브엘리먼트 중에서 반드시 한 개의 서브엘리먼트만이 존재해야 한다는 의미를 포함하고 있다. 따라서 Descriptor 테이블에 대하여 Insert, Delete, Update 연산을 적용하는 경우 문제가 발생하지 않도록 트리거 문장을 적용하여 서브엘리먼트간의 선택적 의미정보를 보존해야 한다.

4. 저장 알고리즘

DIDL 문서를 저장하기 위한 알고리즘을 pseudo-code 형태로 표현하면 그림 7과 같다.

제안하는 저장 알고리즘은 DIDL 문서를 메모리에 적재한 후, DFS 알고리즘에 따라 순회하면서 각 노드를 처리하게 되는데, 자식 노드를 가진 경우에는 함수를 재귀적으로 호출하는 방식을 이용하여 다단계 깊이의 정도에 관계없이 간결하게 처리하였다.

```

Algorithm StoreDIDL_RDB (input:DIDL document)
do
    Initializing local variables
    Initializing array values for sub-element
    do {
        Storing existence information of sub-elements
    } while (sub-element)
    Storing edge information
    do {
        Storing information of attributes
    
```

```

} while (attributes)
if (node.name=="STATEMENT")
    Storing information of values to the BLOB field
if (current_node.HasChildNodes) {
    while (current_node) {
        for (all sub-elements of current_node) {
            if (current_node == reference_node)
                Call StoreDIDL_RDB ();
        }
        current_node=current_node.NextSibling;
    }
    current_node=original_node;
}
} while (DIDL elements)
    
```

그림 7. 저장을 위한 순회 알고리즘

5. 실행 화면

구현 시스템의 수행 모습은 그림 8과 같다. DIDL 문서를 구성하는 각각의 엘리먼트에 해당하는 테이블들과 이들 간의 관계를 표현한 관계테이블의 모습을 모니터링 할 수 있으며, 테이블을 조작하기 위한 다양한 기능의 버튼이 배치되어 있다.

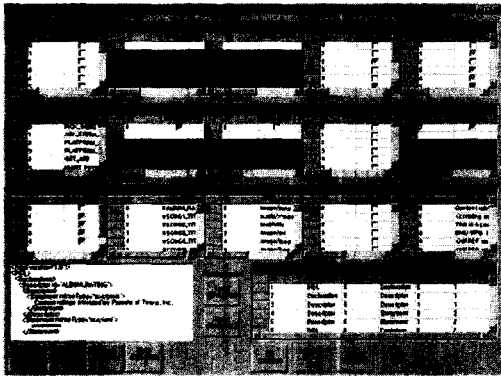


그림 8. 사용자 인터페이스

6. 저장 테이블

그림 9의 간단한 DIDL문서를 이용하여 관계형 데이터베이스에 저장했을 때 실제로 테이블에 매핑되는 모습을 그림 10의 (a), (b), (c) 테이블들을 통하여 나타내고 있다.

```

<DIDL>
  <ITEM>
    <DESCRIPTOR> </DESCRIPTOR>
    <DESCRIPTOR> </DESCRIPTOR>
    <COMPONENT> </COMPONENT>
  </ITEM>
</DIDL>
    
```

그림 9. DIDL 문서

(a) 관계테이블

id	pType	pID	cType	cID	order
1	DIDL	di1	Item	il	1
2	Item	il	Descriptor	de1	1
3	Item	il	Descriptor	de2	2
4	Item	il	Component	c1	3

(b) DIDL 테이블

id	Declarations	Container	Item
D1	Null	Null	Exist

(c) Item 테이블

Id	Condition	Descriptor	Choice	Reference	Item	Component	Annotation
il	Null	Exist	Null	Null	Null	Exist	Null

그림 10. 저장 테이블

7. 질의 예제

관계형 데이터베이스에 저장된 디지털 아이템에 대한 질의는 Digital Item 기반의 질의 형태로부터 XPath를 통하여 이루어지고, 질의내용은 내부적으로 질의 변환기를 통하여 그림 11과 같이 SQL 형태로 변환되어 처리된다.

```

질의:COMPONENT[@id="Movie1"]/RESOURCE[@ref]
SELECT r.ref
FROM Component c, Resource r
WHERE (c.id = "Movie1")
AND (c.id = REL.ParentID)
AND (REL.ChildID = r.id)
    
```

그림 11. SQL로 변환된 질의

노드 간의 상하관계 및 속성의 조건을 질의분석기의 규칙에 의하여 WHERE 조건절에 정의한 후, 조건에 부합된 결과를 DIDL Generator를 통하여 DI 기반의 터미널에서 이용할 수 있도록 생성한다.

V. 결론

본 논문에서는 MPEG-21의 Part2에서 표준으로 제정된 DIDL 문서의 관계형 변환모델에 대하여 살펴보았다. 표준에서 제시하고 있는 내용은 디지털 아이템을 XML 스키마 기반의 표기에 대한 것 뿐이므로 이를 데이터베이스에 저장하기 위한 방안이 필요하다. 기존의 많은 연구가 DTD기반의 매핑모델에 기반한 내용이었으나 제안한 모델은 XML 스키마 기반의 변환을 적용하여 엘리먼트, 애트리뷰트, 값, 관계의 형태로 분류하여 관계형 테이블에 매핑

하였다. 구조적 변환뿐만 아니라 XML 문서에 표현된 기수제약조건, 키제약조건, 도메인 무결성, Nullable 등의 의미적 제약조건 또한 보존할 수 있는 방안을 기존의 연구결과를 토대로 확장하여 적용하였다. 마지막으로 제안한 시스템의 구현과정을 설명하였으며, 이를 위한 시스템 구성도, 구현환경, 알고리즘, 저장 테이블, 질의 예제 등을 통하여 구현된 내용을 상세하게 소개하였다. XML 스키마 기반의 DIDL 문서는 디지털 아이템 기반의 특별한 응용에 최적화된 형태이므로 해당 응용 도메인에 최적화된 변환모델을 제시함에 따라 MPEG-21의 사상인 투명성 및 상호운영성을 위한 다양한 형태의 의미 정보를 보존하는 시스템의 구축을 도모하였다. 향후 DIDL 내에서 이용되는 많은 멀티미디어 표현 언어의 추가적인 모델이 포함된 확장된 모델의 변환을 위한 연구가 요구되며, 다양한 의미적 정보의 보존을 위한 추가적인 노력이 필요하다.

참 고 문 헌

[1] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0 (Second Edition)," W3C Recommendation, <http://www.w3.org/TR/2000/REC-xml-20001006>

[2] Alin Deutsch, Mary Fernandez, Dan Suciu, "Storing Semistructured Data with STORED," *Proc. ACM SIGMOD Conf. Management of Data*, June 1999.

[3] Daniela Florescu, Donald Kossmann, "Storing and Querying XML data Using an RDBMS," *IEEE Data Engineering Bulletin* 22, No. 3, 27-34, 1999.

[4] Jayavel Shanmugasundaram, Kristin Tufte, Gang He, "Relational Databases for Querying XML Documents: Limitations and Opportunities," *Proc. 25th International Conference on VLDB*, Edinburgh, Scotland, September 7-10, 1999.

[5] Dongwon Lee, Murali Mani, Wesley W. Chu, "Schema Conversion Methods between XML and Relations Models," *Knowledge Transformation for the Semantic Web*, IOS Press, Amsterdam, 2003.

[6] XML Schema Part 0: Primer and Part 1:

Structures and Part 2: Datatypes, *W3C Recommendation*, May 2, 2001.

[7] 김정섭, 박창원, 정진완, "XML 스키마를 위한 관계형 스키마 자동생성기의 개관," *한국정보과학회 춘계학술발표논문집*, 제29권, 제1호, pp. 10-12, 2002.

[8] Philip Bohannon, Juliana Freire, Prasan Roy, Jerome Simeon, "From XML Schema to Relations: A Cost-based Approach to XML Storage," *ICDE 2002*.

[9] Jan Bormans, Keith Hill, "MPEG-21 Overview v.5," Shanghai version, Oct. 2002.

[10] Vaughn Iverson, Todd Schwartz, Young-Won Song, Rik Van de Walle, Ernesto Santos, Doim Chang, "MPEG-21 Digital Item Declaration FDIS," Fairfax version, May 2002.

[11] 송정석, 김우생, "MPEG-21의 디지털 아이템 데이터와 관계형 데이터베이스간의 사상모델에 관한 연구," *한국정보처리학회 춘계학술발표논문집*, 제10권, 제1호, pp. 1539-1542, 2003.

[12] J. E. Funderburk, G. Kiernan, J. Shanmugasundaram, E. Shekita, C. Wei, "XTABLES: Bridging relational Technology and XML," *IBM SYSTEMS JOURNAL*, Vol. 41, No. 4, pp. 616-641, 2002.

[13] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts," Fourth Edition, *McGraw-Hill*, 2001.

송 정 석(Jung-suk Song) 정희원

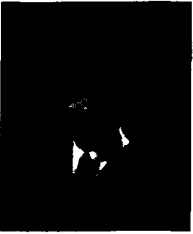
1991년 2월 : 세종대학교 지구
과학과 졸업
1995년 2월 : 광운대학교 전자
계산학과 석사
1998년 8월 : 광운대학교 전자
계산학과 박사과정 수료
2001년 10월~현재 : 한화S&C

기술연구소

<관심분야> 멀티미디어, 데이터베이스, XML, MPEG-21

김 우 생(Woo-saeng Kim)

정회원



1982년 : 서울대학교 수료

1985년 : University of Texas
at Austin 전산학과 졸업

1987년 : University of
Minnesota 전산학과 석사

1991년 : University of
Minnesota 전산학과 박사

1991년 : University of Minnesota Post Doctor

1987년~1988년 : 현대전자. Zeus Computer 과장

2000년~2001년 : 미션텔레콤(주) 이사

2001년 : UC 버클리 대학 교환 교수

1992년~현재 : 광운대학교 컴퓨터공학부 교수

<관심분야> 멀티미디어 시스템, 데이터베이스, 영
상/비디오 처리 및 패턴인식