

MPEG-21의 디지털 아이템 데이터와 관계형 데이터베이스간의 사상모델에 관한 연구

송정석^o 김우생

한화S&C 기술연구소, 광운대학교 컴퓨터공학부
 songjs@hanwha.co.kr^o, kws@cs.kwangwoon.ac.kr

A study on mapping model between digital item data in MPEG-21 and data in relational database

Jungsuk Song^o Woosaeng Kim
 Hanwha S&C, Kwangwoon University

요 약

MPEG-21의 Digital Item은 표준으로 정해진 DID(Digital Item Declaration) 모델을 이용하여 표현된다. 해당 모델은 XML 스키마에 기반하여 정의된 것으로서 뛰어난 표현력과 확장성을 제공할 수 있다. 하지만 정의된 데이터의 저장이나 검색 및 기존 데이터와의 연계에 관한 언급은 표준에서 이루어지지 않고 있다. 본 연구에서는 MPEG-21에서 DID 모델을 이용하여 정의되어 있는 Digital Item을 관계형 데이터베이스에 저장하여 사용할 수 있는 방법에 관하여 논하고자 한다. DID를 이용하여 정의된 내용을 관계형 데이터베이스의 스키마로 매핑하였으며, SQL 언어를 이용하여 기존 데이터와의 연동 검색을 지원하도록 설계하였다.

1. 서론

최근 디지털 시대의 도래에 따른 급변하는 사용자의 요구사항들이 증가함에 따라 디지털 미디어의 생성, 획득, 저장, 검색, 처리 등에 관련된 장치들이 개발되고 있으며 이를 처리하기 위한 각종 멀티미디어 기술이 연구되고 있다. 이에 따라 멀티미디어 콘텐츠 또한 중요한 역할을 담당하게 되면서 콘텐츠의 생성, 공급, 거래, 전송, 관리, 보호, 인증 등을 위한 다양한 요소기술들이 개발되어 왔다.

그러나 수많은 장치와 네트워크의 종류를 함께 사용하는 환경에서 이들 장치와 네트워크에 의존하지 않고 공통적인 시각을 유지하면서 기술을 사용할 수 있는 환경의 필요성이 절실하다. 더불어 이들 간의 상호호환성을 제공하고, 효율적이며 저비용으로 멀티미디어 시스템을 구축할 수 있는 멀티미디어 프레임워크에 해당하는 통합 기반은 제공되지 못하고 있는 실정이다.

이러한 멀티미디어 데이터의 처리를 위한 각종 기반 기술을 제공하기 위하여 ISO/IEC JTC1/SC29/WG11로 표현되는 MPEG 그룹에서는 국제 표준 기술을 정의하여 전세계의 시장에서 공통적으로 사용할 수 있도록 표준화 작업을 진행 중에 있다. 특히 21세기 멀티미디어 프레임워크에 관한 국제 표준 기술을 정의하기 위한 표준화 작업을 'MPEG-21'이라 명명하며 이를 위한 파트별 세부 기술을 분류하여 정의해 나가고 있다[1].

본 논문에서는 MPEG-21의 기본 단위인 디지털 아이템을 선언하는 DID[2] 표현의 관계형 데이터베이스로의 저장에 대하여 다루고자 한다. MPEG-21의 기본 단위인 디지털 아이템은 XML 스키마[3]에 기반한 고유의 정의 모델을 사용하여 표현되는데, XML에 기반한 문법을 채택하고 있기 때문에 다양한 장점을 제공할 것으로 기대된다. MPEG-21에서는 DID를 사용하여 선언하는 방법에 대해서는 표준으로 정의하고 있으나, 선언된 데이터의 저장이나 검색 및 기존 관계형 데이터와의 호환 문제에 대한 언급을 별도로 하고 있지 않다. 이러한

문제의 해결을 위한 기초연구로서 DID로 표현된 디지털 아이템을 관계형 데이터베이스에 저장하여 기존 관계형 데이터와 연동하여 사용하기 위한 방안을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 MPEG-21의 Part2에 정의된 DID에 관련된 표준사항을 살펴보고, 3장에서는 제안하고자 하는 사상모델에 대한 기법 및 질의적용 예제를 설명하며, 4장에서는 결론 및 향후 연구 과제를 제시하고자 한다.

2. 관련연구

2.1. 디지털 아이템

디지털 아이템은 MPEG-21에서 처리되는 개체의 기본적인 단위로서 멀티미디어 콘텐츠와 같은 개념으로 사용된다. 이는 일종의 구조적인 멀티미디어 객체로서 멀티미디어 리소스 자체를 나타내는 물론이고, 디지털 아이템의 식별자와 기술자 등이 담긴 메타데이터 및 디지털 아이템의 구조 또한 포함하는 것으로서 그림 1과 같은 형태로서 하나의 기본적인 개체를 표현한다.

Digital Item = resources + metadata+ structure

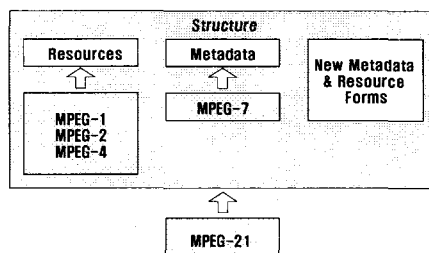


그림 1. MPEG-21의 디지털 아이템 구성방법

2.2. 디지털 아이템의 선언

MPEG-21의 Part 2에 정의되어 있는 DID는 디지털 아이템 선언 모델로서 디지털 아이템이 정형화되고, 유연성을 제공하며, 상호 호환이 가능하도록 설계된 XML 스키마 기반의 모델이다. 디지털 아이템은 묶여 있는 형태에 따라서 다른 요소들이 패키징 되어 있는 Container와 Item으로 우선 나눌 수 있다. Container는 Container와 Item들이 묶여 있는 것이고, Item은 다시 Sub-Item과 Component로 구성된다. 그림 2는 DID를 통하여 표현할 수 있는 디지털 아이템 선언 모델의 계층적 구조도로서 해당 아이템 내에 여러 가지 요소들이 어떠한 식으로 구성되어 하나의 디지털 아이템을 이루는가에 대하여 나타내고 있다.

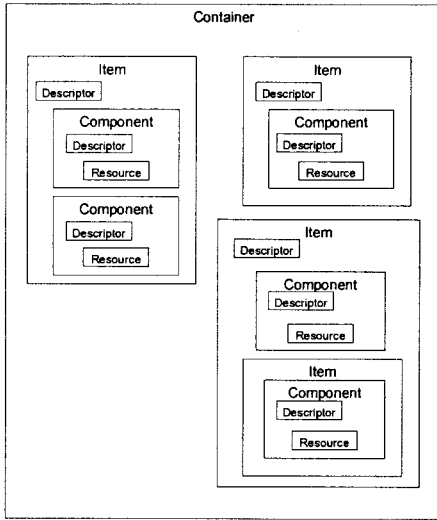


그림 2. DID 모델 예제

2.3. 디지털 아이템 선언 모델

MPEG-21에서는 디지털 아이템의 선언을 위하여 EBNF (Extended Backus-Naur Form) 표기법 및 XML 스키마를 사용하여 각 요소들을 정의하고 있다.

2.3.1 EBNF 표기법을 이용한 DID 모델

그림 3은 EBNF 표기법을 사용하여 표현한 디지털 아이템 선언을 위한 각 요소들의 모습이다.

```

anchor ::= fragment descriptor* condition*
annotation ::= assertion* descriptor* anchor*
assertion ::= predicate*
choice ::= selection* descriptor* condition*
component ::= resource descriptor* anchor* condition*
condition ::= predicate+
container ::= container* item* descriptor*
descriptor ::= descriptor* (component | statement) condition*
item ::= (item | component)* choice* descriptor* condition* annotation*
selection ::= predicate descriptor* condition*
    
```

그림 3. EBNF를 이용한 디지털 아이템 선언 모델

2.3.2 XML 스키마를 이용한 DID 모델

디지털 아이템의 선언을 위한 또 하나의 방법은 XML 스키마를 사용하여 디지털 아이템을 선언하는 방법이다. DID 모델을 구성하는 여러 가지 요소 중에서 Digital Item을 직접적으로 표현하는 Item 요소를 선언하기 위한 XML 스키마를 소개 하자면 다음과 같다.

```

<xsd:element name="Item">
  <xsd:complexType>
    <xsd:sequence>

      <xsd:element ref="Condition" minOccurs="0"
        maxOccurs="unbounded"/>

      <xsd:element ref="Descriptor" minOccurs="0"
        maxOccurs="unbounded"/>

      <xsd:element ref="Choice" minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:choice>
        <xsd:element ref="Reference"/>
        <xsd:choice minOccurs="0"
          maxOccurs="unbounded">
          <xsd:element ref="Item"/>
          <xsd:element ref="Component"/>
        </xsd:choice>
      </xsd:choice>

      <xsd:element ref="Annotation" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>

    <xsd:attributeGroup ref="ID_ATTRS"/>
  </xsd:complexType>
</xsd:element>
    
```

그림 4. Item 요소의 XML 스키마 표현

Item 요소는 사용 가능한 Sub Item 혹은 Component의 모임이다. Descriptor 요소에 의하여 Item에 대한 서술정보가 기술되고, Condition 요소에 의하여 선택 정보가 표시되며, Choice 요소에 의하여 사용자가 별도로 정의한 구성정보로 재구성된다. 부가적인 정보는 Annotation 요소에 의하여 나타낼 수 있다. Item 요소 이외의 나머지 요소들의 표현과 의미 또한 XML 스키마를 이용하여 잘 정의되어 있으며, MPEG-21 Part2인 DID를 설명한 문서를 통하여 자세히 참고할 수 있다 [2].

3. DID 모델의 관계형 모델 적용

본 논문에서는 디지털 아이템을 구성하는 각종 요소들을 관계형 데이터베이스에 저장하여 관계형 데이터베이스에 구축되어 있는 시스템과 자연스럽게 연동할 수 있는 사상기법을 제안한다.

3.1 디지털아이템의 관계형데이터베이스 스키마 매핑

제안하는 기법은 디지털 아이템의 구성을 위한 각 요소들을 관계형 데이터베이스에 하나의 테이블에 대응하여 생성하는 방식이다. 그림 5는 DID에서 사용되는 요소들을 위한 관계형 데이터베이스 스키마를 보여주고 있다.

Anchor (id, Condition, Descriptor, Reference, p_name, p_id)
Annotation (id, Reference, Assertion, Descriptor, Anchor, p_name, p_id)
Assertion (id, assertion, p_name, p_id)
Choice (id, Condition, Descriptor, Selection, p_name, p_id)
Component (id, Condition, Descriptor, Reference, Resource, Anchor, p_name, p_id)
Condition (id, condition, p_name, p_id)
Container (id, Descriptor, Reference, Container, Item, p_name, p_id)
Declarations (id, Item, Descriptor, Component, Annotation, Anchor, p_name, p_id)
Descriptor (id, Condition, Descriptor, Reference, Component, Statement, p_name, p_id)
Didl (id, Declarations, Container, Item)
Item (id, Condition, Descriptor, Choice, Reference, Item, Component, Annotation, p_name, p_id)
Reference (id, reference, p_name, p_id)
Resource (id, resource, p_name, p_id)
Selection (id, Condition, Descriptor, p_name, p_id)
Statement (id, statement, p_name, p_id)

그림 5. DID 요소의 저장을 위한 RDB 스키마

즉 DID를 구성하는 요소들마다 하나씩의 테이블을 생성하고, 각 요소들을 구성하는 Sub-element들을 테이블을 구성하는 하나의 필드에 대응하는 방식이다. 예를 들어 Didl 요소를 구성하는 Sub-element는 Declarations, Container, Item이고, Anchor 요소를 구성하는 Sub-element는 Condition, Descriptor, Reference이다.

Didl 테이블은 디지털 아이템을 구성하는 최상단 요소에 해당하지만 다른 모든 테이블의 경우에는 부모요소를 가질 수 있다. 자식요소와 부모요소 간의 연관관계를 표현하기 위하여 p_name에는 부모요소의 형태(테이블 이름)가 저장되고, p_id에는 부모요소의 고유한 id값이 저장된다. 단 Didl 테이블은 부모요소가 없는 루트요소이므로 p_name, p_id 필드가 불필요하다.

DID의 요소 중에서 일부는 부모 요소로서 2개 이상의 다른 형태의 요소가 나타날 수 있다. 예를 들어 Condition 요소는 Anchor, Choice, Component, Descriptor, Item, Selection 요소의 자식 요소가 될 수 있다.

그림 6은 Condition 테이블의 예제를 보여주는 것으로서 p_name에 부모 요소의 형태를 저장하고 있음을 알 수 있다. p_name은 [4]에서 이용하는 'flag'와 비슷한 용도로 이용할 수 있는데, 예를 들어 부모 요소의 형태가 Anchor라는 사실을 안다면 Anchor 테이블 만을 Condition 테이블과 조인하여 질의하게 되고, 더불어 p_name이 'Anchor'인 튜플 만을 조인에 참여시킬 수 있기 때문에 연산의 부담을 덜 수 있다.

id	condition	p_name	p_id
Condition_id_1	c1	Anchor	Anchor_id_1
Condition_id_2	c2	Choice	Choice_id_1
Condition_id_3	c3	Component	Component_id_1
Condition_id_4	c4	Descriptor	Descriptor_id_1
Condition_id_5	c5	Item	Item_id_1
Condition_id_6	c6	Selection	Selection_id_1

그림 6. Condition 테이블 예제

3.2 디지털 아이템의 테이블 내의 저장

그림 7은 [2]에서 이용된 예제중의 하나로서 영화 '터미네이터' 내용을 담고 있는 디지털 아이템을 DID 표현을 사용하여 정의한 내용이다.

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:01-DIDL-NS">
  <Item id="MOVIE_Item1">
    <Descriptor>
      <Statement mimeType="text/plain"> The Terminator. </Statement>
    </Descriptor>
    ...
    <Component id="MyMovie1-MPEG2-MPML">
      <Condition require="HIGH_SPEED MPEG2_FORMAT"/>
      <Resource ref="movie1-mpg2-mpml.mpg"
mimeType="video/mpeg"/>
    </Component>
    <Component id="MyMovie1-MPEG4-SPL2">
      <Condition require="LOW_SPEED MPEG4_FORMAT"/>
      <Resource ref="movie1-mpg4-spl2.asf"
mimeType="video/x-la-asf"/>
    </Component>
    <Component id="MyMovie1-MPEG4-MPL2">
      <Condition require="MED_SPEED MPEG4_FORMAT"/>
      <Resource ref="movie1-mpg4-mp12.asf"
mimeType="video/x-la-asf"/>
    </Component>
    <Component id="MyMovie1-QTIME">
      <Condition require="LOW_SPEED QTIME_FORMAT"/>
      <Resource ref="movie1.mov" mimeType="video/qtime"/>
    </Component>
  </Item>
</DIDL>
```

그림 7. DID 표현 예제

해당 예제를 테이블에 저장한 모습은 다음과 같은 형태로 나타난다. 각 요소의 고유한 값을 위한 id를 부여하고, 자식 노드가 존재하면 해당 자식 노드의 종류에 해당하는 필드에 'exist' 값을 할당하고, 자식 노드로서 나타나지 않은 나머지 요소를 위한 필드에는 'null' 값을 할당한다. 예를 들어 Didl 요소의 자식 요소로는 Declarations, Container, Item 요소가 포함될 수 있는데, 예제에서는 Didl 요소의 자식 요소로서 Item 요소만을 가지고 있으므로 Item 필드에는 exist를 할당하고, Declarations와 Container 필드에는 null 값을 할당한다.

DIDL 테이블

id	Declarations	Container	Item
Didl_id_1	null	null	exist

Item 테이블

id	Condi- tion	Descr- iptor	Choic- e	Refer- ence	Item	Comp- onent	Annot- ation	p_na- me	p_id
Movie- Item1	null	exist	null	null	null	exist	null	Didl	Didl- id_1

Component 테이블

id	Condi- tion	Descr- iptor	Referen- ce	Resourc- e	Anc- hor	p_na- me	p_id
MyMovie1-M- PEG2-MPML	exist	null	null	exist	null	Item	MOVIE- Item1
MyMovie1-M- PEG4-MPL2	exist	null	null	exist	null	Item	MOVIE- Item1
MyMovie1-M- PEG4-QTIME	exist	null	null	exist	null	Item	MOVIE- Item1
MyMovie1-M- PEG4-SPL2	exist	null	null	exist	null	Item	MOVIE- Item1

3.3. 디지털 아이템을 이용한 질의

관계형 데이터베이스에 저장된 디지털 아이템의 내용을 이용하여 정보를 검색하는 예제를 SQL 질의를 사용하여 처리해 보도록 한다.

질의 1. Reference 요소의 ref 속성이 'mpg4'를 포함하는 Reference 요소의 ref 속성과 Component 요소의 id 속성을 검색하라.

```
SELECT Component.id, Resource.ref
FROM Component, Resource
WHERE Resource.p_id=Component.id and
Resource.ref Like '%mpg4%'
```

결과 집합

Component.id	Reference.ref
MyMovie1-MPEG4-SPL2	movie1-mpg4-spl2.asf
MyMovie1-MPEG4-MPL2	movie1-mpg4-mpl2.asf

질의 2. 이름이 "User2"인 사용자의 name, telephone, address 필드를 검색하고, 보유하고 있는 디지털 아이템 내용 중에서 Item 요소의 id 속성과, Statement 요소의 statement 속성을 검색하라.

```
SELECT User.name, User.telephone, User.address,
Item.id, Statement.statement
FROM User, REL_User_DidI, DidI, Item, Descriptor,
Statement
WHERE User.id = REL_User_DidI.UserID AND
REL_User_DidI.DidIID = DidI.id AND DidI.id =
Item.F_KEY AND Item.id = Descriptor.F_KEY AND
Descriptor.Statement = Statement.id AND User.Name =
'User2'
```

결과 집합

User.name	User.telephone	User.address	Item.id	Statement.statement
User2	222-2222	월계동 1-2	MOVIE_Item1	The Terminator.

질의 2에서 사용한 User 테이블은 기존의 관계형 데이터베이스에서 사용하던 일반적인 사용자의 정보를 저장한 테이블이고, REL_User_DidI 테이블은 User 테이블의 ID와 DidI 테이블의 ID 정보를 가진 관계테이블로서 기존의 관계형 테이블과 디지털 아이템 테이블을 상호 연계하여 검색할 수 있음을 알 수 있다.

4. 결론 및 향후 연구 과제

본 논문에서는 MPEG-21 기반의 DID로 표현된 데이터를 관계형 데이터베이스에 저장하여 기존의 데이터와 연동하여 사용할 수 있는 방법에 대하여 알아보았다.

제안하는 방법은 DID로 표현된 데이터를 DID Schema의 규칙성을 유지하면서 관계형 데이터베이스에 적용할 수 있는 방법이다. 따라서 기존의 관계형 데이터베이스 시스템, SQL 질의어, 프로그래밍 언어 등을 모두 수용하면서 최신 기술로 표현된 데이터를 사용할 수 있다는 장점을 지니게 된다. 또한 제한된 요소를 사용하여 정의하는 DID 스키마를 기반으로 하는 모델이기 때문에 테이블의 개수가 정해져 있고, 순환참조의 가능성이 낮기 때문에 상대적으로 조인 연산이 작게 소모되고, 테이블 관리가 단순하다는 점이 또 하나의 장점으로 작용될 수 있다.

향후 연구과제로는 해당 모델의 다양한 응용으로의 적용을 위한 사상모델의 추가확장을 들 수 있다.

5. 참고문헌

- [1] J. Bormans, K. Hill, "MPEG-21 Overview v.5," Shanghai version, October 2002.
- [2] V. Iverson, et al, "MPEG-21 Digital Item Declaration FDIS," Fairfax version, May 2002.
- [3] "XML Schema Part 1: Structures and Part 2: Datatypes," W3C Recommendation, 2 May 2001.
- [4] D. Florescu and D. Kossmann, "Storing and querying XML data using an RDBMS," IEEE Data Eng. Bull., 22(3):27-34, 1999.
- [5] J. Shanmugasundaram, et al, "Relational Databases for Querying XML Documents: Limitations and Opportunities," Proc. of the VLDB '99, pp. 302-314, Sep., 1999.
- [6] J. Fong, et al, "Converting relational database into XML document," Proc. 12th International Workshop on Database and Expert Systems Applications, pp. 61-65, 2001.
- [7] A. Silberschatz, et al, "Database System Concepts," 4th edition, McGrawHill.
- [8] 안영희, 황부현, "멀티미디어 데이터에 대한 XML문서 저장 관리 시스템 설계 및 구현," 제18회 한국정보처리학회 춘계학술발표대회 논문집, 제9권, 제2호, 2002.