

MPEG-21 디지털 아이템 정의 모델

LG전자기술원 송영원 · 장도임

1. 서 론

다양한 유무선 네트워크를 통한 멀티미디어 콘텐츠의 생성, 공급, 거래, 관리, 보호 및 이용 등과 관련된 기술적 특성은 매우 다양하게 발전하고 있고, 그 시장 규모는 향후 수 년 내에 기하급수적으로 증가하리라 예상된다. 그러나, 오늘날 이러한 멀티미디어 콘텐츠의 생성부터 이용에 관한 멀티미디어 객체의 취급 단위에 관한 디지털 아이템 정의 모델(MPEG-21에서는 이를 Digital Item Declaration이라 함)에 관한 융통성(flexible) 있고 규칙성(uniform) 있으며 상호 호환적인(interoperable) 객체 모델 또는 그 프레임워크는 거의 없다[1]. 따라서, 이러한 통합적이고, 보편적인 모델이 없이, 전자 상거래 비즈니스 모델에 나타난 다양한 User들(디지털 아이템을 취급하는 생성자로부터 소비자까지의 모든 entities)이 제각기 다른 기술적, 경제적 관점에 따라 각각의 모델을 이용한다면, 상호 호환성의 부재로 콘텐츠 유통과 관련된 인터넷 시장의 큰 혼란이 예상된다[2]. 그러므로, 이러한 문제 해결을 위한 상호 호환적인 국제적 디지털 아이템 정의 모델 기술의 개발은 매우 중요하다.

본 고는 멀티미디어 콘텐츠의 생성부터 이용을 위한 통합적인 프레임워크에 관한 국제 기술 표준인 MPEG(Moving Picture Experts Group)-21에서 정의한 요소기술들 중의 하나인 디지털 아이템 정의 모델에 관한 것이다. 디지털 아이템 정의 모델의 목표는 융통성 있고, 규칙성 있으며, 상호 호환적인 디지털 아이템(멀티미디어 콘텐츠 취급단위)을 정의하는 것이다. 여기서, 첫째, 융통성 있는 모델이란, 멀티미디어 타입(예를 들어, 정지영상, 동영상, 음성, 텍스트, 소프트웨어, 웹페이지 등)에 관계없이 정의된 모델이 적용될 수 있어야 함을 의미한다. 둘째, 규칙성 있는

모델이란, 미래에 새로운 타입의 멀티미디어 콘텐츠가 이용될 경우, MPEG-21에서 제안된 이 정의 모델이 일관성 있게 확장될 수 있어야 한다. 그리고 상호 호환적 모델이란, 생성된 멀티미디어 콘텐츠의 포맷이나 이를 전송, 이용하는 네트워크나 디스플레이 터미널의 프로토콜의 종류에 의존하지 않고 호환적으로 응용될 수 있어야 함을 의미한다. 현재, MPEG-21에서 제안되고 있는 디지털 아이템 정의 모델은 Working Draft Version 2.0(도큐먼트 번호: ISO/IEC 21000-2)이고 그 제목은 "Information Technology- Multimedia Framework -Part2: Digital Item Declaration"이다[3].

본 고는 이 도큐먼트의 기술적 내용을 중심으로 설명한다. MPEG-21의 전체 기술을 규정하고 있는 기술보고(도큐먼트 번호: ISO/IEC 21000-1, "Information Technology-Multimedia Framework-Part1: Vision, Technologies and Strategy")에 설명되어 있듯이, 디지털 아이템은 구조적인 디지털 형태의 멀티미디어 객체로 개념적으로 그 리소스와 메타데이터로 구성되어 있다[1]. 이 객체는 MPEG-21 멀티미디어 프레임워크의 기본 취급 단위로 정의하며 멀티미디어 타입이나 포맷 등과 무관하게 응용될 수 있는 추상적 모델이다.

정의된 디지털 아이템은 아이템(item)과 컨테이너(container)라는 두 가지 요소들로 선언되어져 있다. 먼저 아이템은 다른 서브 아이템(sub-items)이나 컴포넌트(components)에 의해 정의되어질 수 있는데, 여기서 컴포넌트는 개념적으로 가장 작은 규모의 리소스와 메타데이터로 구성되어져 있다. 컨테이너(container)는 서브 컨테이너(sub-containers)나 아이템(item)에 의해 패키지 형태로 구성될 수 있는 좀 더 규모가 큰 형태의 디지털 아이템이다. 여기서,

아이템과 컨테이너의 가장 큰 차이는 아이템은 디지털 아이템 이용자의 터미널이나 네트워크 환경 등에 맞춰서 원하는 디지털 아이템의 조건들을 선택할 수 있는 구성 조건들(configuration choices)로 정의된다는 것이다. 컨테이너는 그룹핑의 목적으로 논리적인 패키지나 카테고리 특성을 갖도록 정의된다. 디지털 아이템 정의 모델은 디지털 아이템이 XML 기반으로 인스턴스(혹은 도큐먼트)될 수 있도록 정의되어 있다[4].

본 고의 2장에서는 BNF(Backus-Naur Form)를 이용한 디지털 아이템의 의미 모델(semantic model)을 기술하고 그 정의된 모델 요소들의 개념들을 설명한다[5]. 3장에서는 XML-Schema를 이용하여 디지털 아이템의 구조 모델(syntactic model)과 디지털 아이템에 관한 상세 구조를 표현한다[6].

2. 멀티미디어 객체 모델

일반적으로, 멀티미디어 객체는 그 규모에 있어 그림 1과 같이 두 가지 형태(atomic Digital Item, package Digital Item)로 구분되어질 수 있다, 즉, 더 이상 쪼개어질 수 없는 원자적 수준의 객체(atomic or primitive level object) (예를 들어, 하나의 정지 영상이나 최소한의 의미 있는 음악 데이터 등)와 그 원자적 수준의 객체들을 묶은 형태의 복합적 수준의 객체(composite level object) (예를 들어, 정지 영상들을 담고 있는 사진 앨범이나 여러 음악 곡을 담고 있는 CD 등)로 구성될 수 있다. 그런데 만약 어떤 패키지 된 복합적 수준의 객체들이 컨테이너에 담긴다면 (예를 들어, 한 편의 동영상과 여러 정지 영

상 및 음악 곡들이 담긴 비디오 뮤직), 이것 역시, 좀 더 큰 규모이긴 하나, 복합적 수준의 객체에 속한다.

그러나 멀티미디어 콘텐츠를 전송, 이용하는 다양한 네트워크와 터미널 환경을 고려할 때, 복합적 수준의 객체는 다시 두 가지 형태로 구분될 수 있다. 즉, 하나는 멀티미디어 이용자가 그 콘텐츠 이용과 관련하여 자신이 갖추고 있는 터미널과 네트워크 환경에 적합하게 콘텐츠를 구성할 수 있는(configurable) 조건들을 포함하는 객체와 단순히 객체의 묶음 형태로 존재할 수 있는 두 가지 경우이다. 그림 1에서 아이템(item)이 원하는 조건으로 멀티미디어 구성할 수 있는 조건들이 포함된 객체이고, 컨테이너(container)는 그런 선택의 조건이 필요하지 않은 것이다.

3. BNF에 의한 디지털 아이템 정의 모델

본 장에서는 디지털 아이템 정의 모델에서 선언 혹은 정의된 요소들과 그들이 갖고 있는 개념들을 BNF를 이용하여 표현하고 이를 설명한다.

이 장에서 이탤릭(*italic*)체로 표시된 요소들은 이 모델 안에서 정의된 것이다. 상자 안의 BNF 표현은 모델 안에서의 요소들 사이의 정확한 관계 또는 구조를 나타낸다.

3.1 Container

*container*는 *items*와 *containers*가 묶인 구조이다. 이들 *items*와 *container*들의 묶음은 적절한 공급 및 이용을 위한 논리적인 패키지와 카테고리(혹은 shelf 개념)의 특징을 갖도록 정의한다. *descriptor*는 묶음의 목적에 맞는 정보를 *container*의 “이름표”로 붙일 수 있게 한다(예를 들면, 패키지에는 배달 지시를, 또는 카테고리에는 분류 범주를 기술한다).

container 자체는 *item*이 될 수 없다.

container ::= container* item* descriptor*

3.2 Item

*item*은 개념적으로 *sub-items*와 *components*의 묶음 형태이다. *item*은 (*item* | *component*)뿐만 아니라, 내용을 설명하는 *descriptor*와 적절한 이용 조건으로 구성할 수 있는 선택 정보를 담은 *choice*, 그리고 그 이외의 부가적 정보를 정의할 수 있는 *annotation*과 같은 메타데이터들로 정의된다.

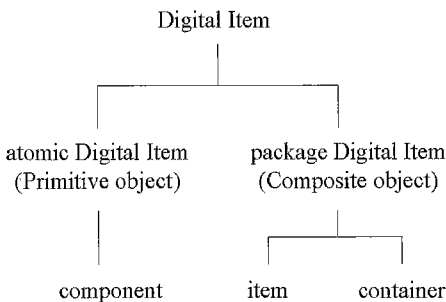


그림 1 디지털 아이템(멀티미디어 객체)의 분류: 규모에 의한 분류

```
item ::= (item | component)+ choice*
        descriptor* condition* annotation*
```

3.3 Component

*component*는 *item*의 구성하는 요소이다. 이는 의미 있는 디지털 아이템을 구성하기 위한 부품과 같은 것으로 URI(Uniform Resource Identifier) 형식 [7]의 레퍼런스를 갖는 *resource*와 이 *resource*를 설명하는 *descriptor*, 그리고 *resource*의 일부 범위를 연결시킬 수 있는 *anchor*로 정의된다. *component*의 *condition*은 정의된 *component*의 이용과 관련하여 보통 *component*가 이용되는 터미널과 네트워크에 관한 기술적 조건으로 기술된다.

```
component ::= resource descriptor* anchor*
            condition*
```

3.4 Anchor

*anchor*는 배의 닻이 배가 어느 특정 위치에 정박할 수 있게 해주는 역할을 하듯이, *resource* 내의 어느 범위를 지정할 수 있는 기능을 한다. 이 기능을 *fragment*에서 기술하고, *descriptor*는 *anchor*의 내용을 설명한다.

```
anchor ::= fragment descriptor* condition*
```

3.5 Descriptor

*descriptor*는 이것을 포함하는 요소의 내용을 설명한다. 이 설명은 *icon*과 같은 *thumbnail*(이는 일종의 *component*이다)이나, 작품의 제목과 작자와 같은 텍스트 형태의 *statement*로 될 수 있다. 이 *descriptor*는 회귀적(recursive)으로 정의되므로 *descriptor* 내에 또 다른 *descriptor*가 계층적으로 포함될 수 있다.

```
descriptor ::= descriptor* (component |
                        statement) condition*
```

3.6 Condition

*condition*은 이것을 포함한 요소가 선택적임을 말하는 것이며, 이 요소를 포함할 것인지 혹은 유효하게

할 것인지를 결정한다. 어떤 요소의 포함 여부를 결정할 때, *condition* 안에 있는 여러 개의 *predicate*는 논리곱(AND)으로 결합되고, 한 요소 안에 있는 여러 개의 *condition*은 논리합(OR)으로 결합된다. 또한 *predicate*에는 부정(NOT)을 붙일 수 있다.

```
condition ::= predicate+
```

3.7 Choice

*choice*는 하나의 *item*을 구성하는데 영향을 줄 수 있는 일련의 *selection*들을 기술한다. *choice*안에 있는 *selection*들은 꼭 한 개만 선택되거나(exclusive) 임의 개수가 선택(inclusive, 전부 선택하거나 또는 선택이 없는 경우를 포함하는)될 수 있다.

```
choice ::= selection+ descriptor* condition*
```

3.8 Selection

*selection*은 *item* 내부에 있는 하나 또는 그 이상의 *condition*에 영향을 주는 특정한 결정을 기술한다. 만약 *selection*이 선택되면, 이것의 *predicate*가 참이 되고 만약 선택되지 않으면, 이것의 *predicate*는 거짓이 된다. 만약 결정되지 않은 상태로 남으면, 이것의 *predicate*는 미결정이 된다.

```
selection ::= predicate descriptor* condition*
```

3.9 Annotation

*annotation*은 디지털 아이템에서 정의된 요소들 중 ID type을 가지고 있는 요소에 그 요소의 내용과 구조를 변경시키지 않고 추가적인 정보(*assertions*, *descriptor*, *anchor*)를 덧붙일 때 사용한다.

```
annotation ::= assertion* descriptor* anchor*
```

3.10 Assertion

*assertion*은 *choice*내의 *selection*과 연결된 몇 개의 *predicates*를 참, 거짓 또는 미결정으로 정함으로써 전부 또는 부분적으로 *choice*의 구성을 정한다.

assertion ::= predicate*

3.11 Resource

*resource*는 멀티미디어 콘텐츠의 내용에 해당하는 것으로 예를 들어, 정지/동영상, 오디오 클립, 텍스트 형태의 문서, 실행 가능한 소프트웨어 혹은 패키지, 웹 페이지 등과 같은 것이다. 모든 *resource*는 MPEG-21 디지털 아이템 정의 모델에서 URI reference 형식으로 명확한 주소에 의해 위치를 파악할 수 있어야 한다.

3.12 Fragment

*fragment*는 *resource* 안의 특정 위치 또는 범위를 정확히 나타낸다. *fragment*는 *resource*의 표현 내에 단편 식별자(#)가 있는 경우에 이용될 수 있다.

3.13 Statement

*statement*는 디지털 아이템 내에서 정의된 여러 요소들(*container*, *item*, *component*, *anchor*, *choice*, *annotation* 등)의 내용을 텍스트 형태로 표현하는 것이다.

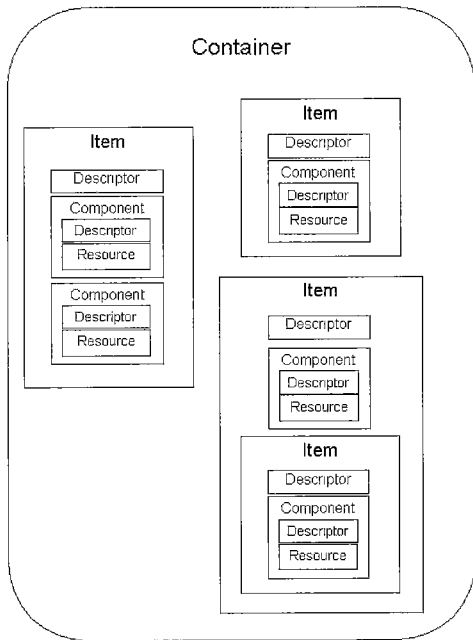


그림 2 디지털 아이템의 예

현하는 것이다.

3.14 Predicate

*predicate*는 참, 거짓 또는 미결정의 구별이 명확한 선언(Declaration)이다.

그림 2는 디지털 아이템 모델의 계층적 구조로서 모델 안의 여러 요소들이 어떠한 관계가 있는지를 보여준다.

4. XML-Schema에 의한 디지털 아이템 정의 모델

이 장에서는 3장의 BNF에 의한 디지털 아이템 모델이 XML-Schema에 의해 정의된 내용을 기술한다. 이 디지털 아이템 모델은 MPEG-21 Digital Item Declaration Working Draft에서 DIDL (Digital Item Declaration Language)이라고 불리어진다.

DIDL 문서는 하나의 ITEM 요소 또는 하나의 CONTAINER 요소를 가지는 DIDL 루트 요소로 구성된다.

또한, DIDL은 BNF에 의한 디지털 아이템 정의에 기술되지 않은 REFERENCE, DECLARATIONS, OVERRIDE 요소를 정의한다. 이들 요소들은 DIDL 내에서 특별한 목적으로 사용된다.

DIDL은 XML의 ID 특성 타입을 많이 사용한다. 일반적으로, 이 타입의 특성은 하나의 DIDL 요소와 다른 요소 사이의 내부 연결에 사용된다. 예를 들면, 많은 DIDL 요소들은 'ID' 특성을 갖는데, 이로써 REFERENCE 요소의 대상이 될 수 있고, ANNOTATION 요소에 의한 주석이 가능해진다. 이름이 'ID'가 아니지만, ID 타입의 특성을 사용하는 경우도 있는데, SELECTION 요소의 SELECT_ID 특성은 특정 SELECTION들과 CONDITION을 연결하는데 사용되고, CHOICE_ID 특성은 ASSERTION 요소와 특정 CHOICE들을 연결하는데 사용된다. DIDL 문서가 병합될 때 충돌을 피하기 위해 ID 값의 변경이 필요할 수도 있다.

4.1 DIDL

DIDL은 디지털 아이템 인스턴스 문서의 루트 요소로, DECLARATIONS를 선택적으로 포함하고,

하나의 CONTAINER 또는 하나의 ITEM을 반드시 포함한다.

```
<xsd:element name="DIDL">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="DECLARATIONS"
minOccurs="0"/>
      <xsd:choice>
        <xsd:element ref="CONTAINER"/>
        <xsd:element ref="ITEM"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="CONTAINER">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="DESCRIPTOR"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:choice>
        <xsd:element ref="REFERENCE"/>
        <xsd:sequence>
          <xsd:element ref="CONTAINER"
minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element ref="ITEM"
minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attributeGroup ref="ID_ATTRS"/>
  </xsd:complexType>
</xsd:element>
```

4.2 DECLARATIONS

DECLARATIONS는 프로그램 언어에서 헤더 파일로 선언되어진 것과 유사한 역할을 한다. DIDL 문서 내에서 정의된 여러 요소들 중 ITEM, DESCRIPTOR, COMPONENT, RESOURCE, ANNOTATION 요소들이 여러 번 이용되어질 경우, DECLARATIONS 내에 선언된 요소들을 참조하여 이용한다.

```
<xsd:element name="DECLARATIONS">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="ITEM"/>
      <xsd:element ref="DESCRIPTOR"/>
      <xsd:element ref="COMPONENT"/>
      <xsd:element ref="RESOURCE"/>
      <xsd:element ref="ANNOTATION"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

4.3 CONTAINER

CONTAINER는 XML-Schema에서 정의된 complexType/sequence 형태로, DESCRIPTOR와 하나만 선택되어질 수 있는 CONTAINER(이는 정의되어질 CONTAINER의 sub 요소이다), ITEM으로 정의된다. CONTAINER의 attribute는 reference로서 ID_ATTRS를 값으로 갖는데 이는 xsd:ID와 같다.

4.4 ITEM

ITEM은 ITEM을 조건적으로 포함하는데 필요한 CONDITION과 이 ITEM을 설명하는 DESCRIPTOR, 선택적 구성에 필요한 CHOICE, ITEM의 기본 구성 요소인 COMPONENT 또는 sub ITEM 혹은 다른 ITEM을 가리킬 수 있는 REFERENCE로 정의된다. 이미 정의된 ITEM을 변경시키지 않고 추가 정보를 넣을 필요가 있을 경우에 대비해 ANNOTATION을 정의한다.

```
<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:choice>
        <xsd:element ref="REFERENCE"/>
        <xsd:choice minOccurs= 0
maxOccurs="unbounded">
          <xsd:element ref="CHOICE"/>
          <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>
```

4.5 COMPONENT

COMPONENT는 이 COMPONENT를 조건적으로 포함하는데 필요한 CONDITION, resource 내용을 설명하는 DESCRIPTOR, RESOURCE 또는 이미 정의된 resource를 가리키는 REFERENCE와 resource의 특정 범위를 지정하는 ANCHOR로 정의된다.

```
<xsd:element name="COMPONENT">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="DESCRIPTOR"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:choice>
        <xsd:element ref="REFERENCE"/>
        <xsd:element ref="RESOURCE"
minOccurs="1" maxOccurs="unbounded"/>
      </xsd:choice>
      <xsd:element ref="ANCHOR"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="ID_ATTRS"/>
  </xsd:complexType>
</xsd:element>
```

4.6 RESOURCE

RESOURCE는 attribute들로 정의된다.

REF attribute에 실제 resource의 URI를 지정하고, TYPE attribute에는 resource의 타입을 URI로 지정한다. LOCAL_PATH attribute에는 resource의 지역 사본의 위치를 지정한다.

```
<xsd:element name="RESOURCE">
  <xsd:complexType mixed="true">
    <xsd:attribute name="TYPE"
type="xsd:uriReference"/>
    <xsd:attribute name="REF"
type="xsd:uriReference"/>
    <xsd:attribute name="LOCAL_PATH"
type="xsd:uriReference"/>
  </xsd:complexType>
</xsd:element>
```

4.7 DESCRIPTOR

DESCRIPTOR는 이를 조건적으로 포함시킬 수 있게 하기 위한 CONDITION, DESCRIPTOR를 계층적으로 정의하기 위한 하위 수준의 DESCRIPTOR, 그리고 하나만 선택되어질 수 있는 REFERENCE, COMPONENT, STATEMENT로 정의된다.

```
<xsd:element name="DESCRIPTOR">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="DESCRIPTOR"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:choice>
        <xsd:element ref="REFERENCE"/>
        <xsd:element ref="COMPONENT"/>
        <xsd:element ref="STATEMENT"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attributeGroup ref="ID_ATTRS"/>
  </xsd:complexType>
</xsd:element>
```

4.8 STATEMENT

STATEMENT는 보통 텍스트와 기계가 읽을 수 있는 다양한 형식(well-formed XML과 같은)을 포함하여 다른 어떤 데이터 형식도 포함할 수 있다. 데이터 형식은 TYPE attribute에서 URI로 지정된다. 만약 어떤 기관이 그들만의 독특한 형식을 가지고 있다면 그것 또한 유일한 URI로 지정할 수 있다.

STATEMENT에는 사람이 읽기에 편하도록 보

```
<xsd:element name="STATEMENT">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:any namespace="##any"
processContents="skip" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="TYPE"
type="xsd:uriReference"/>
  </xsd:complexType>
</xsd:element>
```

통의 텍스트를 포함할 수도 있고, 응용프로그램을 위해서는 기계가 읽을 수 있는 형식(well-formed XML과 같은)을 포함할 수도 있다. 이때, STATEMENT에 well-formed XML 문장을 포함하도록 하려면, *statement*에 포함된 XML 문장의 문법(즉, XML Schema)이 이름 공간(namespace)으로 지정되어야 한다[8].

4.9 ANCHOR

ANCHOR는 CONDITION, DESCRIPTOR 요소와, 우선 순위를 표현하는 PRECEDENCE, URI와 '#' 문자를 사용하여 *resource* 내의 원하는 위치 또는 범위를 나타내는 FRAGMENT attribute로 정의된다.

```
<xsd:element name="ANCHOR">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="DESCRIPTOR"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="PRECEDENCE"
      type="xsd:unsignedInt"
      use="default" value="0"/>
    <xsd:attribute name="FRAGMENT"
      type="xsd:uriReference"/>
    <xsd:attribute name="NAME"
      type="xsd:ID" />
  </xsd:complexType>
</xsd:element>
```

4.10 CHOICE

CHOICE는 CONDITION, DESCRIPTOR, SELECTION, OVERRIDE 요소들과, MIN_SELECTIONS, MAX_SELECTIONS, DEFAULT, CHOICE_ID attribute들로 정의된다.

MIN_SELECTIONS와 MAX_SELECTIONS는 *choice*에서 반드시 정해야 하는 SELECTION 수를 지정한다. 예를 들어, MIN_SELECTIONS과 MAX_SELECTIONS이 생략되면 SELECTION을 몇 개를 해도 상관없다는 뜻이고, MIN_SELECTIONS과 MAX_SELECTIONS 특성이 모두 '1' 이면, 반드시 하나의 SELECTION을 선

택해야 한다는 뜻이다.

CHOICE는 OVERRIDE를 포함함으로써, 현재 문서 또는 외부 문서 안에 있는 하나 또는 그 이상의 다른 CHOICE를 무시할 수 있다. 각 OVERRIDE는 현재 CHOICE가 대신할 다른 CHOICE를 지정한다.

```
<xsd:element name="CHOICE">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="DESCRIPTOR"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="SELECTION"
maxOccurs="unbounded"/>
      <xsd:element ref="OVERRIDE"
minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="MIN_SELECTIONS"
      type="xsd:nonNegativeInteger"/>
    <xsd:attribute name="MAX_SELECTIONS"
      type="xsd:nonNegativeInteger"/>
    <xsd:attribute name="DEFAULT"
      type="xsd:IDREFS"/>
    <xsd:attribute name="CHOICE_ID"
      type="xsd:ID"/>
  </xsd:complexType>
</xsd:element>
```

4.11 SELECTION

SELECTION은 CHOICE 내에 나열되어 있는 선택사항이다. SELECT_ID attribute와 그 값은 BNF에 의한 디지털 아이템 정의 모델의 *predicate*와 동일한 역할을 한다.

SELECTION과 CHOICE는 CONDITION을 갖기 때문에 모두 조건적으로 제어되어질 수 있다. 예를 들어, DIDL 문서가 비디오 클립의 포함 여부를 선택하는 CHOICE와 비디오 클립의 포맷을 선택하는 CHOICE를 갖는다고 가정하자. 포맷을 선택하는 CHOICE는 '비디오 클립'이라는 CONDITION을 갖는다. ITEM을 구성할 때, '비디오 클립'을 선택하지 않으면 CONDITION에 의해 포맷을 선택하는 CHOICE는 무시된다.

SELECTION은 OVERRIDE를 포함함으로써, 현재 문서 또는 외부 문서 안에 있는 하나 이상의 다른 SELECTION을 무시할 수 있다.

```

<xsd:element name="SELECTION">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="DESCRIPTOR"
minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="OVERRIDE"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="SELECT_ID"
type="xsd:ID" use="required"/>
  </xsd:complexType>
</xsd:element>

```

4.12 CONDITION

CONDITION은 어떤 요소가 *predicate* 비교를 통해 조건적으로 포함될 필요가 있을 때 사용한다. 조건을 만족하기 위해서는 REQUIRE attribute와 그 값에 의해 결정되는 *predicate*들은 반드시 참이 되어야 하고, EXCEPT attribute와 관련된 *predicate*들은 반드시 거짓이 되어야 한다. CONDITION에서 attribute 값은 IDREFS로 표현되며, SELECTION 요소의 SELECT_ID attribute에 나타난 값과 비교하기 위하여 그 값을 참조한다.

```

<xsd:element name="CONDITION">
  <xsd:complexType>
    <xsd:attribute name="REQUIRE"
type="xsd:IDREFS"/>
    <xsd:attribute name="EXCEPT"
type="xsd:IDREFS"/>
  </xsd:complexType>
</xsd:element>

```

4.13 OVERRIDE

OVERRIDE 요소는 부모 요소에 의해 대체될 CHOICE 또는 SELECTION을 지정하는데 사용된다.

TARGET attribute는 현재 문서 또는 외부 문서에 있는 무시될 CHOICE 또는 SELECTION을 지정한다.

```

<xsd:element name="OVERRIDE">
  <xsd:complexType>
    <xsd:attribute name="TARGET"
type="xsd:uriReference"/>
  </xsd:complexType>
</xsd:element>

```

4.14 REFERENCE

REFERENCE는 DIDL 내에서 정의된 여러 요소들(CONTAINER, ITEM, COMPONENT, DESCRIPTOR, ANCHOR, 또는 ANNOTATION)을 참조할 수 있다. 이는 외부의 DIDL 문서의 요소를 참조하는 외부 참조와 내부 문서 안에 있는 요소를 참조하는 내부 참조 모두 가능하다. 외부 참조는 다음과 같이 표현될 수 있다.

http://abc.com/document1.didl#element_id 내부 참조는 다음과 같이 표현될 수 있다.
#element_id

```

<xsd:element name="REFERENCE">
  <xsd:complexType>
    <xsd:attribute name="URI"
type="xsd:uriReference" use="required"/>
  </xsd:complexType>
</xsd:element>

```

4.15 ANNOTATION

ANNOTATION 요소는 ITEM 또는 CONTAINER의 원본에 영향을 주지 않고, DESCRIPTOR와 ANCHOR 요소들을 논리적으로 추가하는데 사용된다. 예를 들어, 사용자가 책갈피를 연결하거나, 전자 서명된 작품에 서명을 무효로 하지 않고 주석을 다는데 사용된다. CHOICE의 SELECTION들에 *predicate* 값을 정하는 ASSERTION도 추가할 수도 있다.

```

<xsd:element name="ANNOTATION">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="REFERENCE"/>
      <xsd:sequence>
        <xsd:element ref="ASSERTION"
minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="DESCRIPTOR"
minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="ANCHOR"
minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:choice>
    <xsd:attribute name="TARGET"
type="xsd:IDREF" use="required"/>
    <xsd:attributeGroup ref="ID_ATTRS"/>
  </xsd:complexType>
</xsd:element>

```


4.16 ASSERTION

ASSERTION은 TARGET attribute에 지정한 CHOICE 안에 있는 predicate들을 참 또는 거짓으로 정하는데 사용된다. TRUE attribute에는 참으로 만들 SELECTION들의 SELECT_ID를, FALSE attribute에는 거짓으로 만들 SELECTION들의 SELECT_ID를 써준다.

```
<xsd:element name="ASSERTION">
  <xsd:complexType>
    <xsd:attribute name="TARGET"
type="xsd:IDREF" use="required"/>
    <xsd:attribute name="TRUE"
type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="FALSE"
type="xsd:IDREFS" use="optional"/>
  </xsd:complexType>
</xsd:element>
```

5. DIDL 문서 예제

다음 예는 디지털 아이템을 XML로 작성한 것이다. DIDL은 하나의 ITEM으로 구성되어 있는데, 이 ITEM은 크게 하나의 CHOICE와 두 개의 COMPONENT로 이루어지며, MP3와 WMA 형식의 오디오 중에 하나를 사용자가 선택하여 구성할 수 있는 디지털 아이템이다. DESCRIPTOR들이 CHOICE와 SELECTION 요소의 내용을 설명하고 있다.

```
<DIDL>
  <ITEM>
    <CHOICE MIN_SELECTIONS=" 1"
MAX_SELECTIONS=" 1" >
      <DESCRIPTOR>
        <STATEMENT TYPE="/text/plain " >
          What format would you prefer?
        </STATEMENT>
      </DESCRIPTOR>
      <SELECTION SELECT_ID=" MP3_FORMAT">
        <DESCRIPTOR>
          <STATEMENT TYPE="/text/plain" > I want
MP3</STATEMENT>
        </DESCRIPTOR>
      </SELECTION>
      <SELECTION SELECT_ID=" WMA_FORMAT" >
```

```
<DESCRIPTOR>
  <STATEMENT TYPE="/text/plain" > I want
WMA</STATEMENT>
</DESCRIPTOR>
</SELECTION>
</CHOICE>
<COMPONENT>
  <CONDITION REQUIRE=" MP3_FORMAT" />
  <RESOURCE REF=" clip.mp3"
TYPE=" /audio/mp3" />
</COMPONENT>
<COMPONENT>
  <CONDITION REQUIRE=" WMA_FORMAT" />
  <RESOURCE REF=" clip.wma"
TYPE=" /audio/wma" />
</COMPONENT>
</ITEM>
</DIDL>
```

6. 맺음말

본 고에서는 MPEG-21에서 제안한 디지털 아이템 정의 모델의 필요성과 기본적인 요구 조건들(융통성, 규칙성, 상호 호환성)을 서론에서 논하였다. 일반적인 멀티미디어 객체 모델에 관한 근본적인 접근 방법을 2장에서 기술하였다. 3장에서는 이 접근 방법때 따른 구체적인 디지털 아이템 정의 모델의 구성 요소들을 BNF를 이용하여 기술하였다. 4장에서는 XML-Schema를 이용한 구조적인 디지털 아이템 정의 모델을 상세하게 기술하였고, 5장에서 그 예제를 보였다.

현재 MPEG-21 디지털 아이템 정의 모델에 관한 기술은 Working Draft Ver. 2.0으로 작성되어 있는데, MPEG에서 수행하는 CE(Core Experiment) 과정을 통하여 더욱 정확하고 정교하며 실용성 있는 표준 모델로 발전되어 나갈 것이다.

참고문헌

[1] ISO/IEC 21000-1, "Information Technology-Multimedia Framework-Part1: Vision, Technologies and Strategy," March 2001.
 [2] Leonardo Chiariglione, "Technologies for E-Content," published on the Internet at <http://www.csel.it/leonardo/paper/wipo99/>

index.htm. August 1999.

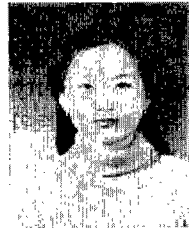
- [3] ISO/IEC 21000-2, "Information Technology-Multimedia Framework-Part2: Digital Item Declaration," March 2001.
- [4] W3C Recommendation, "Extensible Markup Language(XML) 1.0 (2nd Ed)," 6 October 2000.
- [5] ISO/IEC 14977:1996(E), "Backus-Naur Form(BNF)," 1996.
- [6] W3C Proposed Recommendation, "XML Schema Part 1:Structures and Part 2: Datatypes," 16 March 2001.
- [7] IETF Draft Standard, "Uniform Resource Identifier(URI)-RFC 2396," August 1998.
- [8] W3C Recommendations, "Namespaces in XML," 14 January 1999.

송 영 원



1981 한양대학교 전자공학과(학사)
 1984 서울대학교 전자공학과(석사)
 1997 Iowa State Univ. 전기 및 컴퓨터공학과(박사)
 1984~현재 LG 전자기술원, 정보기술연구소
 관심분야:통신 및 신호처리, MPEG-21, Metadata Framework, VLSI 회로 설계, 비파괴 평가 (NDE)
 E-mail:ysong@LG-Elite.com

장 도 임



1998 이화여자대학교 컴퓨터학과(학사)
 2000 이화여자대학교 컴퓨터학과(석사)
 2000~현재 LG 전자기술원, 정보기술연구소
 관심분야:MPEG-21, 메타데이터 모델링, 실시간 모니터링, 디지털 리포팅
 E-mail:doim@LG-Elite.com

● 2001년 병렬 및 분산 시스템에 관한 국제학술회의 ●

- 일 자 : 2001년 6월 26~29일
- 장 소 : 경주
- 내 용 : 논문발표 등
- 문 의 처 : 고려대학교 컴퓨터학과 박명순 교수

Tel. 02-3290-3193

<http://www.icpads2001.org>