

<http://dx.doi.org/10.7236/IIBC.2015.15.1.157>

IIBC 2015-1-20

XML 스키마의 개념적 모델링을 위한 확장된 개체관계 모델

Extended Entity-Relationship Model for Conceptual Modeling of XML Schema

정인환*, 김영웅**

In-Hwan Jung*, Young-Ung Kim**

요약 XML은 인터넷 상의 데이터의 표현 및 교환의 사실상 표준으로 자리 잡고 있으며, XML 자체가 데이터를 저장하기 위한 논리적 구조를 표현하고 있지만, 다양하고 복잡한 표현으로 인해 문서 구조를 한 눈에 파악하기에는 어려운 점이 있어 개념적 모델의 도구로 사용하기 적절하지 못한 점이 있다. 본 논문은 XML 스키마 문서 구조를 확장된 개체관계 모델을 이용하여 그래픽 형태로 개념적 모델링할 수 있는 기법을 제안한다. 이를 위해 XML 스키마 구조를 표현하기 위해 확장된 개체관계 모델을 제시하고, XML 스키마 요소들을 확장된 개체관계 모델로 표현하기 위한 표현 규칙들을 제시하고, 제안한 기법으로 모델링한 개념적 모델을 논리적 모델인 XML 스키마 문서로 매핑하는 사례를 통해 본 기법의 완전성을 보여준다.

Abstract XML has become one of the most influential standard language for representing and exchanging data on internet. However, XML itself has a ability to represent a logical structure for storing and managing data, it is inadequate to use as a conceptual modeling tool because of its complexity for representing the document structures. In this paper, we propose the graphical form of conceptual modeling techniques for representing the structure of the XML schema documents using an extended entity relationship diagram. For this, extended entity relationship model is presented for representing the XML schema structure, transformation rules are presented for transforming extended entity relationship model into XML schema document to show the completeness of the proposed model.

Key Words : XML Schema, Diagrammatical Representation, Extended Entity Relationship Diagram

1. 서론

일반적으로 다이어그램은 기호, 선, 점 등을 사용해 각종 사상의 상호 관계나 과정, 구조 등의 의미를 빠르고 정확하게 전달하는 시각 언어으로써 개념적 설계 단계에서 개발환경에 독립적으로 실세계의 현상을 분석하고, 설계

하는데 유용한 수단으로 사용되고 있다. 개념적 설계 단계에서는 관리 대상의 객체, 그 객체들의 특성, 그들 간의 관계 등 현실 세계가 내포하는 의미들을 모두 포함한다. 대표적인 개념적 데이터 모델인 개체관계 모델(Entity-Relationship Model)은 기본적으로 개체집합(entity set), 개체의 속성들(attributes), 개체 간의 관계집

*정희원, 한성대학교 컴퓨터공학과

**정희원, 한성대학교 컴퓨터공학과(교신저자)

접수일자 : 2014년 10월 27일, 수정완료 : 2014년 12월 14일

게재확정일자 : 2015년 2월 13일

Received: 27 October, 2014 / Revised: 14 December, 2014

Accepted: 13 February, 2015

**Corresponding Author: yukim@hansung.ac.kr

Dept. of Computer Eng., Hansung University, Korea

합(relationship set)들을 명시하여 현실 세계를 개념적으로 표현하는 기법이다^[1].

한편, XML(eXtensible Markup Language)은 W3C(World Wide Web Consortium)에서 제안하였으며, 웹문서를 구조화하는 사실상의(de facto) 표준 형식언어로서 XML 자체가 논리적 데이터 모델로써 데이터베이스 응용에 널리 사용되고 있다^[2].

일반적으로 데이터베이스 관점에서 실세계의 데이터를 설계할 때 개념적(conceptual), 논리적(logical), 물리적(physical) 설계 단계를 거친다. 이 기준에서 XML은 논리적 모델에 속한다고 볼 수 있다. 따라서 XML 문서를 모델링할 때 먼저 실세계의 현상을 표현하기 위해 개념적 설계를 한 후 이를 바탕으로 XML로 변환한다. 개념적 데이터베이스 설계 단계에서 사용되는 개체관계모델, UML(Unified Modeling Language)^[3], ORM(Object Relational Mapping)^[4] 등은 XML의 특성을 표현하는 데는 차이가 있어 자연스럽게 변환이 되지 않는다^[5].

본 논문은 XML 특성을 개념적 설계 단계에서 표현하기 위해 기존의 개체관계 모델이 지원하지 않는 개념을 수용하는 확장된 개체관계 모델을 제시하고, 이를 기반으로 XML 스키마를 구성하는 각 요소들을 확장된 개체관계 다이어그램으로 표현하기 위한 표현 규칙들을 제시하고, 제안한 기법으로 모델링한 결과를 생성 규칙과 생성 알고리즘 통해 XML 스키마 정의(XSD: XML Schema Definition)로 매핑하는 과정을 보여준다. 개념적 모델은 논리적 모델과는 독립적으로 표현되어야 하지만 XML에서 사용하는 스키마 언어는 DTD, XML Schema, RELAX_NG 등 다양하며, 각각의 언어는 서로 다른 구조와 제약조건을 명시하고 있어 모두를 수용하는 개념적 모델을 표현하기에는 모델의 복잡성으로 인해 현실적으로 어려움이 있다. 따라서 본 논문에서는 XML 스키마를 논리적 모델로 정하고 이를 근거로 개념적 모델을 제시한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 XML 기반의 개념적 모델에 관한 관련 연구를 기술하고, 제 3장에서는 확장된 개체관계 모델에서의 표현규칙을 기술한다. 이 다이어그램으로부터 XML 스키마로 변환하기 위해 요구되는 다이어그램 정보를 저장하기 위한 자료구조를 제시한다. 제 4장에서는 제안한 확장된 개체관계 모델로부터 XML 스키마 문서를 생성하기 위한 자료구조와 생성 규칙 및 생성 알고리즘을 기술하며, 끝으로 제 5장에서 결론 및 향후 연구과제에 대해 기술한다.

II. 관련 연구

XML과 개체관계 모델이 연관된 관련 연구로는 크게 개체관계 모델로부터 XML 문서로 변환하는 기법과, 개체관계 모델을 이용하여 XML을 개념적으로 설계하는 기법이 있다. 전자의 대표적인 연구로는 ER-to-XML^[6], ERX^[7] 등이 있으며, 후자의 대표적인 연구 결과로는 Xer, Xere, ERex 등이 있다.

Xer 기법^[8]은 XML 스키마 개념을 개체관계 모델로 표현하기 위해 기존의 객체관계 모델에서 표현할 수 없는 개념들을 반영한 확장된 개체관계 모델을 제안하였다. 확장된 개념으로는 요소의 순서의 유무(sequence/all), 'mixed'타입의 요소, 요소와 그에 내포된 요소의 표현 등의 개념이 있다. 또한 복합요소가 복합요소를 포함할 경우는 이를 관계로 표현하였고, 관계에 참여하는 개체의 매핑대응수(mapping cardinality)도 표현하고 있다. 개체가 선택적으로 부개체(sub-entity)를 포함할 경우("xs:choice" 경우) 이를 개체 내부에서 중첩된 형태로 표현하고 있다.

Xere 기법^[9]은 정보 중심의 XML 스키마 문서를 입력으로 받아 이를 개체관계 모델로 표현하기 위해 매핑 알고리즘을 제안하였다. 이 기법에서는 각 요소들의 컨텐츠 모델(sequence, choice, all 등)에 따라 각각을 개체관계 다이어그램으로 표현하고, 속성 타입에 따라 주기를 선정하고, 요소간의 중첩구조를 관계 집합으로 표현하였다. 매핑 알고리즘은 XSLT를 이용하여 구현하였으며, 이 알고리즘의 건전성(soundness)과 완전성(completeness)에 대해서도 기술하였다.

ERex 기법^[10]은 XML 특성을 표현하기 위해 개체관계 모델에 category, coverage constraints, order constraints 개념을 확장한 개념적 모델을 제안하고 있다. category는 슈퍼 타입의 인스턴스가 서브타입의 인스턴스에 속하지 않는 경우나, 서브타입이 독자적인 주기를 갖는 경우를 고려한 것이며, coverage constraints는 카테고리 관계에 참여하는 서브 타입의 인스턴스의 참여도와 서브타입 내의 인스턴스의 중복 여부에 따라 total coverage와 exclusive coverage로 구분하며, order constraints는 XML 스키마에서의 sequence 컨텐츠 모델을 표현하기 위한 제약조건이다. 이 기법에서는 XGrammar를 정의하여 XML 스키마를 명세하였으며, ERex 스키마를 XGrammar 스키마로 변환하는 알고리

증을 제시하고 있다.

Xer과 Xere 기법은 XML 스키마 문서를 알고리즘을 통해 개체관계 모델로 자연스럽게 표현가능한 장점이 있으나, XML이 갖는 특징을 모두 고려하지 않고 있으며, 변환이 가역적이지 못한 결과로 추출된 개체관계 다이어그램으로부터 원래의 XML 스키마 문서의 구조를 추론할 수 없는 단점을 내포하고 있다.

EReX 기법은 논리적 모델과는 독립적으로 개념적 모델을 제시하고 있다는 점에서는 매우 의미가 있는 연구이나, 연구의 중심이 형식언어 중심으로 되어 있어 개념적 모델이 가져야하는 다이어그램으로 표현하는 기법이 부재하여 가시성 측면에서 떨어지는 문제점을 내포하고 있다.

III. 표현 규칙

이 장에서는 XML 스키마의 개념적 모델을 위해 기존의 개체관계 모델에 XML 특성을 표현하기 위한 개념들을 확장하고, 이를 토대로 각각의 요소들을 표현하는 표현 규칙들을 기술한다. 본 논문에서 기술한 XML 스키마 문서는 단지 주요 관점만을 명세하고 있으며, 확장된 개체관계 모델 또한 변환의 논점이 되는 주요 부분만 기술한다.

1. 확장된 개체관계 모델 기본 개체집합 구조

XML 스키마 모델과 개체관계 모델 사이에는 구조 및 개념의 차이가 있어 두 모델 간에 1:1로 자연스럽게 변환이 되지 않는다. 이를 해결하기 위해 기존의 개체관계 다이어그램을 확장하여 사용한다. 본 논문에서는 데이터 타입이나 도메인 무결성 제약조건 등 개념적 설계에 영향을 주지 않는 요소들에 대해서는 기술을 생략한다. 그림 1은 복합요소(complex element)를 표현하기 위한 개체집합의 구조를 보여준다.

개체집합은 세 개의 영역으로 나누어지며, 상위영역은 요소명과 ① 요소에 적용되는 콘텐츠 모델을 지정하고 ②, 중간영역은 요소 내에서 정의되는 부요소명(③)과 부요소에 적용할 대응수(④)를 지정하고(④), 하위영역은 요소에 속한 속성명(⑤)을 지정한다.

②에 적용되는 콘텐츠 모델은 (a: all, s: sequence, c: choice)가 있으며, ④에 적용되는 대응수는 “?”(zero or

one), “1”(exactly one), “*” (zero or more), “+”(one or more)가 있으며, 대응수가 정의되어있지 않으면 “1”을 적용한다. 속성에는 순서가 없으며, 속성이 키일 경우 속성명에 밑줄을 그어 구분한다.

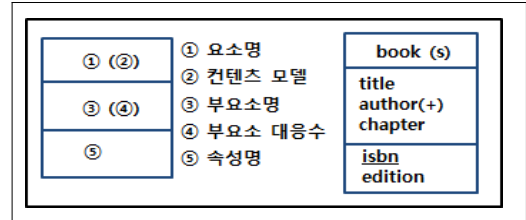
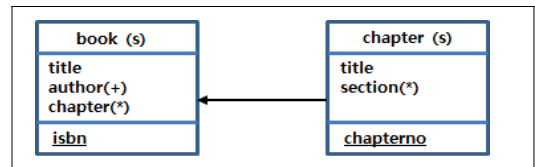


그림 1. 개체집합 기본 구조
 Fig. 1. Basic structure of entity set

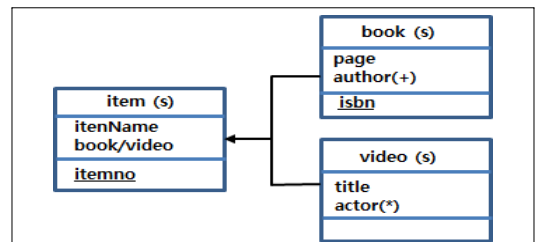
2. 확장된 개체관계 모델의 표현 규칙

[부모요소와 자식요소] 자식요소는 부모요소 내에서 정의되는 하위요소를 말한다. 자식요소는 복합요소 및 단순요소(simple element)가 될 수도 있다. 자식요소가 복합요소일 경우 부모요소와 관계로써 표현한다.



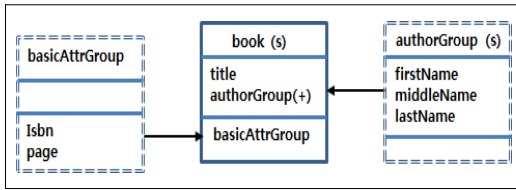
위의 다이어그램에서 요소 *book(s)*는 하위요소들 (*title*, *author*, *chapter*)이 위에서 아래의 순서를 가짐을 의미한다. 속성 *isbn*이 key로 선언되었으므로 XML 스키마 변환시 타입을 ID로 지정됨을 의미하며, 요소 *chapter*는 요소 *book*의 자식요소로서 부모요소와 자식요소와의 관계를 설정하기 위해 두 요소를 선으로 연결하며 부모요소 쪽을 화살표로 표시한다.

[선택 요소] 부모요소 내의 자식요소가 선택적으로 (choice) 선언될 경우 자식요소를 묶어 부모요소와 연결하며 부모요소 쪽을 화살표로 표시한다.



위의 다이어그램에서 부모요소 *item*은 자식요소 *itemName*, *book*, *video*를 순차적으로 정의하고, 이 중 *book*, *video*를 선택적으로 선언한 경우이다. 이 경우 선택적 요소명들을 "/"를 이용해서 하나로 묶고 관계 연결은 자식요소들을 묶어 부모요소와 하나의 실선으로 연결한다.

[그룹] 그룹은 다수의 요소/속성들을 하나의 단위로 묶어 재사용 가능한 콘텐츠를 사용하기 위해 정의한다. 그룹에는 요소그룹(element group)과 속성그룹(attribute group)이 있다

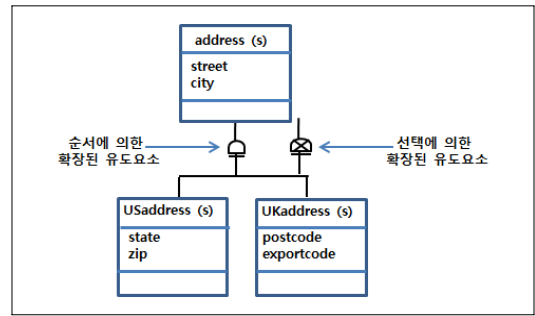


위의 다이어그램에서 *authorGroup*은 *book* 요소 내의 자식요소에서 요소 그룹으로 정의되어 있으며, 테두리를 이중 점선으로 표시하여 복합요소와 구분한다.

*basicAttrGroup*은 *book*에서 속성그룹으로 선언되었으며, 속성 그룹은 그룹요소와 동일하게 테두리를 이중 점선으로 표시하고, 요소 그룹은 속성 영역이 비워있는 것에 반해, 속성 그룹은 자식요소 영역이 비워있어 이 둘을 구분한다.

[확장된 유도요소] 확장된 유도요소(deriving element by extension)는 순서에 의한 유도요소(extension in sequence)와 선택에 의한 유도요소(extension in choice)가 있다. 순서에 의한 유도요소는 기본요소와 유도요소 간에 중첩된 일반화(generalization with overlapping) 개념으로 표현한다. 즉, 기본 요소의 모든 자식 요소가 유도 요소의 자식 요소로 계승됨을 의미한다. 선택에 의한 유도요소는 기본 요소와 유도 요소 간에 분리된 일반화(generalization with disjoint) 개념으로 표현된다. 즉, 기본 타입의 모든 요소가 유도 타입의 요소가 선택적으로 계승됨을 의미한다.

다음의 다이어그램에서 요소 *shipTo*는 자식요소로 *address*를 가지며, *address* 요소는 자식 요소로써 *street*와 *city*를 가지며, *address* 요소의 유도 요소로서 *USaddress*와 *UKaddress*가 선언되었음을 나타내고 있다. 선택에 의한 확장된 유도 요소와의 구분은 기본 요소와 유도 요소와의 연결선 사이에 상이한 기호로써 구분한다.



IV. XML 스키마 매핑

본 장에서는 제 3장에서 개념적 모델로 제시한 확장된 개체관계 다이어그램으로 개념적 설계를 마치면 이를 토대로 논리적 모델인 XML 스키마로 매핑하는 과정을 기술한다. 개념적-논리적 매핑을 위해 먼저 확장된 개체관계 다이어그램 정보를 저장하는 자료구조를 정의하고, 이 자료구조를 근거로 XML 스키마를 매핑하기 위한 매핑규칙과 알고리즘을 간략히 기술한다.

1. 매핑을 위한 자료 구조

확장된 개체관계 다이어그램으로부터 XML 스키마 문서로 변환하기 위해서는 다이어그램 정보를 저장하는 자료구조를 정의하여야 한다. 그림 2는 확장된 개체관계 다이어그램 정보 저장을 위한 자료구조를 보여준다.

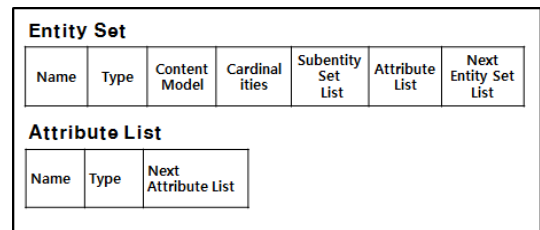


그림 2. 자료구조
Fig. 2. Data structures

그림 2에서 Entity Set 내의 Name은 요소명을 저장하고, Type은 복합요소(CE), 단순요소(SE), 요소그룹(EG), 속성그룹(AG), 확장(EB), 축소(RB), 타입(T)으로 구분하고, Content Model은 순서(S), 무순서(A), 선택(C)으로 구분한다. Cardinality는 1, *, +로 구분하고, Subentity Set List는 이 요소의 자식 요소들을 연결하는

링크정보이며, Attribute List는 이 요소에서 정의되는 속성들을 연결하는 링크정보이고, Next Entity Set List는 이 요소의 다음 요소를 연결하는 링크정보이다. Attribute List 내의 Name은 속성명을 저장하고, Type은 속성의 데이터 타입을 저장하고, Next Attribute List는 이 속성의 다음 속성을 연결하는 링크정보이다.

2. XML 스키마 매핑

본 절에서는 제 3장에서 기술한 표현규칙을 적용하여 다이어그램으로 표현한 확장된 개체관계 모델을 논리적 모델인 XML 스키마 문서로 변환하는 과정을 보여준다. 그림 3은 도서와 비디오 주문처리를 확장된 개체관계 다이어그램으로 표현한 개념적 설계 결과를 보여준다.

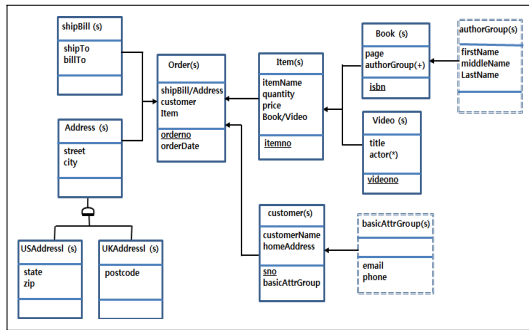


그림 3. 확장된 개체관계 다이어그램
 Fig. 3. Extended Entity-Relationship Diagram

그림 3에서 *shipBill/Address* 및 *Book/Video*는 선택적 요소로 표현하였고, *Address*와 *USAddress* 및 *UKAddress*와의 관계는 순서에 의한 확장된 유도요소로 표현하였고, *email*과 *phone*을 속성그룹으로 묶어 표현하였다.

그림 4는 그림 3의 확장된 개체관계 다이어그램으로부터 정보를 추출하여 저장한 자료구조를 시각적으로 표현한 것이다. 데이터 타입은 속성이 키일 경우 ID로, 그 이외의 경우 string으로 선언한다.

그림 4의 자료구조로부터 제 3장에서 기술한 표현규칙을 적용하여 XML 스키마 문서를 생성하기 위한 생성 규칙과 생성 알고리즘은 다음과 같다. XML 스키마 문서 생성은 표현하는 방식에 따라 다양하게 나타날 수 있으나, 본 논문에서는 적용한 생성 규칙은 다음과 같다.

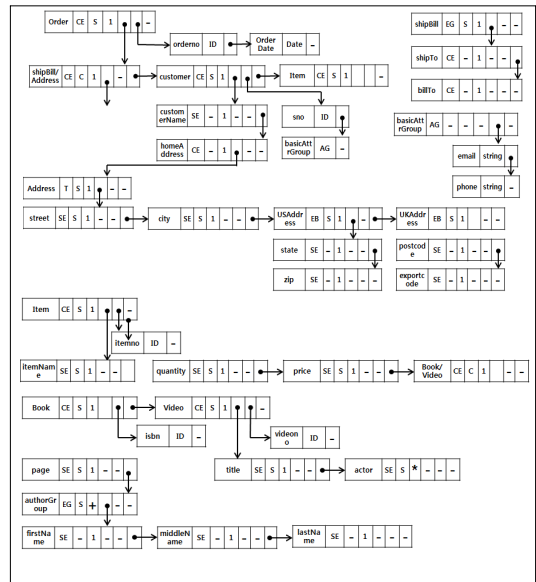


그림 4. 그림 5로부터 추출한 자료구조
 Fig. 4. Data structures extracted from Fig 5

[XML 스키마 생성 규칙]

- 단순요소 및 속성은 부모요소에 내포하여(inlining) 생성한다.
- 복합요소는 별도의 타입을 선언하여 생성한다. 이때 타입명은 요소명 뒤에 Type을 붙여 사용한다.
- 그룹은 별도의 Type으로 선언하고, 명칭은 요소(속성)명을 그대로 사용한다.
- 속성이 키일 경우 ID를, 그이외의 그 이외의 경우 string으로 변환한다.
- 유도요소는 일반요소와 동일하게 적용하며 <xs:extension base>로 선언해서 생성한다.

[XML 스키마 생성 알고리즘]

1. 자신의 부모요소가 없는 요소를 찾아 큐에 저장한다.
2. 큐로부터 하나씩 읽어내어 다음을 반복한다.
 - 2.1 Type이 EG, AG, T일 경우 스키마를 생성한다.
 - 2.2 Type이 CE일 경우 요소명, 타입, 컨텐츠 모델을 생성하고 Subentity Set List를 하나씩 읽어내어 다음을 반복한다.
 - 2.2.1 Type이 CE이면 요소명, 타입, 컨텐츠 모델을 생성한 다음 큐에 저장한다.
 - 2.2.2 Type이 SE이면 요소명, 타입만 생성한다.

2.2.3 Type이 EB이면 요소명, extension base="부
모타입명"으로 생성한다.

2.2.4 Attribute List를 하나씩 읽어내어 속성명, 타
입을 생성한다.

3. 큐가 empty일 때까지 2를 반복한다.

그림 5는 그림 3의 확장된 개체관계 다이어그램으로
부터 생성 규칙과 생성 알고리즘을 적용해 생성한 XML
스키마 문서를 보여준다.

```
<xs:group name="authorGroup">
  <xs:sequence>
    <xs:element firstName type="xs:string"/>
    <xs:element middleName type="xs:string"/>
    <xs:element lastName type="xs:string"/>
  </xs:sequence>
</xs:group>
<xs:attributeGroup name="basicAttrGroup">
  <xs:attribute name="email" type="xs:string"/>
  <xs:attribute name="phone" type="xs:string"/>
</xs:attributeGroup>
<xs:element name="Order" type="OrderType"/>
<xs:complexType name="OrderType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="shipBill" type="shipBillType"/>
      <xs:element name="Address" type="AddressType"/>
    </xs:choice>
    <xs:element name="customer" type="customerType"/>
    <xs:element name="Item" type="ItemType"/>
  </xs:sequence>
  <xs:attribute name="orderno" type="xs:ID"/>
  <xs:attribute name="orderDate" type="xs:string"/>
</xs:complexType>
<xs:complexType name="shipBillType">
  <xs:sequence>
    <xs:element name="shipTo" type="xs:string"/>
    <xs:element name="billTo" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AddressType">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="street" type="xs:string"/>
    <xs:element name="city" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="USAAddress">
  <xs:complexContent>
    <xs:extension base="AddressType">
      <xs:sequence>
        <xs:element name="state" type="xs:string"/>
        <xs:element name="zip" type="xs:integer"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="UKAddress">
  <xs:complexContent>
    <xs:extension base="AddressType">
      <xs:sequence>
        <xs:element name="postcode" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="customerType">
  <xs:sequence>
    <xs:element name="customerName" type="xs:string"/>
    <xs:element name="homeAddress" type="AddressType"/>
  </xs:sequence>
</xs:complexType>
```

```
</xs:sequence>
<xs:attribute name="sno" type="xs:ID"/>
<xs:attributeGroup name="basicAttrGroup"/>
</xs:complexType>
<xs:complexType name="ItemType">
  <xs:sequence>
    <xs:element name="itemName" type="xs:string"/>
    <xs:element name="quantity" type="xs:string"/>
    <xs:element name="price" type="xs:string"/>
    <xs:choice>
      <xs:element name="Book" type="BookType"/>
      <xs:element name="Book" type="VideoType"/>
    </xs:choice>
    <xs:attribute name="itemno" type="xs:ID"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BookType">
  <xs:sequence>
    <xs:element name="title"/>
    <xs:element name="authorGroup" minOccurs="1"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="isbn" type="xs:ID"/>
</xs:complexType>
<xs:complexType name="VideoType">
  <xs:sequence>
    <xs:element name="title"/>
    <xs:element name="actor" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="videono" type="xs:ID"/>
</xs:complexType>
```

그림 5. XML 스키마 문서
Fig. 5. XML schema document

V. 결론

본 논문은 XML 스키마 문서 구조를 확장된 개체관계 모델을 이용하여 그래픽 형태로 개념적 모델링할 수 있는 기법을 제안하였다. 이를 위해 XML 스키마 구조를 표현하기 위해 확장된 개체관계 모델을 제시하고, XML 스키마 요소들을 확장된 개체관계 모델로 표현하기 위한 표현 기법들을 제시하고, 제안한 표현 기법을 이용하여 확장된 개체관계 다이어그램으로부터 XML 스키마 문서를 생성하는 사례를 통해 본 기법의 완전성을 보여준다. 생성 과정은 확장된 개체관계 다이어그램으로부터 정보를 추출하여 제안한 자료구조로 저장하고, 이로부터 생성규칙과 생성 알고리즘을 통해 XML 스키마 문서를 생성하였다.

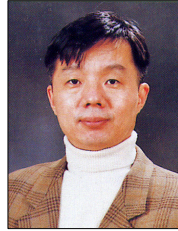
본 연구는 아직 프로토타입 구현 단계로서 향후 계속 진행할 연구 과제로는 크게 직관에 의존한 접근방법을 체계적인 형식적 토대(formal foundation)를 갖추는 것과, 아직 해결하지 못한 부분으로 다양한 XML 개념들을 수용할 수 있는 모델로 확장하는 연구가 진행할 예정이다.

References

- [1] B. Thalheim, Entity-Relationship Modeling : Foundation of Database Technology, Spring Verlag, 2000.
- [2] World Wide Web Consortium, eXtensible Markup Language(XML) 1.0, <http://www.w3.org/TR/1998/REC-xml-19980210>, 1998.
- [3] OMG. "Unified Modeling Language Specifications, version 1.5, 2003. <http://www.uml.org/>.
- [4] T. Halpin, "Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design", Morgan · Kaufmann Pub., 2001.
- [5] M. Necasky, "Conceptual Modeling for XML: A Survey", Proc. of the 6th Annual International Workshop on Databases, Vol 176, 2006.
- [6] S. Jin, W. Kang, "Mapping Rules for ER to XML Using XML Schema", Proc. of the Southern Association for Information System Conference, 2007.
- [7] Y. Kim, "ERX: A Generation Tool of XML Schema based on Entity-Relationship Model", The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol 13, No 2, 2013.
- [8] A. Sengupta, S. Mohan, R. Doshi, "XER-Extensible Entity Relationship Modeling", Proc. of XML 2003, 2003.
- [9] G. Penna, et al., "Towards the Expected Interoperability between XML and ER Diagrams", Technical Report TRCS/G0102, Dept. of Computer Science, Univ. of L'Aquila, 2002.
- [10] M. Mani, "EReX: A Conceptual Model for XML" Proceedings of the 2nd International XML Database Symposium(XSystem 2004), 2004.

저자 소개

김 영 웅(정회원)



- 1993년 : KAIST 전산학과 박사
- 1984 ~ 1997년 : KT 통신망 연구소
- 1997년 ~ 현재 : 한성대학교 컴퓨터 공학과 교수
- <주관심분야 : 데이터모델링, 소프트웨어 신뢰도, 소프트웨어 설계>

정 인 환(정회원)



- 2000년 : KAIST 정보및통신공학과 박사
- 1985년 ~ 1998년 : 삼성전자 시스템사업부 선임연구원
- 2001년 ~ 현재 : 한성대학교 컴퓨터 공학과 교수

<주관심분야 : 분산데이터베이스, 멀티미디어통신>

※ 본 연구는 한성대학교 교내연구비 지원 과제임.