

<https://doi.org/10.7236/IIBC.2017.17.6.211>

IIBC 2017-6-27

개념적 모델로의 XML 스키마의 역변환 기법

Reverse Engineering of XML Schema to Conceptual Model

정인환*, 김영웅*

In-Hwan Jung*, Young-Ung Kim*

요약 XML은 데이터의 표현 및 교환의 사실상 표준으로 자리 잡고 있으며, XML 자체가 데이터를 저장, 관리하기 위한 구조로 되어있지만, 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 보다 쉽게 이해할 수 있도록 개념적 모델로 변환하는 기법을 제안한다. 이를 위해 개념적 모델 도구로 사용한 CMXML을 기술하고, XML 스키마 구조를 CMXML 다이어그램으로 변환하기 위한 변환 규칙과 자료구조를 제시한다. 변환절차는 XML 스키마 문서를 입력받아 본 연구에서 제안하는 변환기법을 적용하여 CMXML 다이어그램을 생성한다.

Abstract XML is becoming a de facto standard for the representation and exchange of data, and XML itself is well structured to store and manage data, but it is difficult to understand the document structure at a glance due to its diverse and complex expressions. Thus, it is not appropriate to use it as a tool of conceptual model. In this paper, we propose a method to transform XML schema document structure into conceptual model. To do this, we describe CMXML as a conceptual modeling tool and present transformation rules and data structures for transforming an XML schema structure into CMXML diagram. In the transformation procedure, the XML schema document is input and the CMXML diagram is generated by applying the transformation rules.

Key Words : XML Schema, CMXML Diagram, Transformation Rule, Reverse Engineering

1. 서론

XML(eXtensible Markup Language)은 데이터를 표현하고 교환하는 사실상의 표준으로 사용되고 있다^[1]. XML 자체가 데이터를 저장, 관리하기 위한 구조로 되어 있지만, 아직도 많은 응용에서는 데이터 저장 및 추출의 용이성으로 인해 XML 데이터를 데이터베이스에 저장하여 관리하고 있는 실정이다.

개념적 설계 단계에서는 관리 대상의 객체, 객체들 간

의 관계 등 현실 세계가 내포하는 의미들을 모두 포함한다. 대표적인 개념적 데이터 모델인 개체 관계 모델이 있지만 XML 스키마 구조는 문서 위주의 표현에 중점을 두고 있는 반면 개체 관계 모델은 개체와 그들 사이의 관계를 표현하는 데 중점을 두고 있기 때문에 두 모델 사이에는 표현의 차이가 있어 자연스러운 변환이 어렵다^[2].

XML의 개념적 설계를 위한 많은 연구가 진행되고 있다. 대표적인 연구로는 개체 관계 모델을 확장한 ER-to-XML^[3], UML을 기반으로 하는 MDD^[4], 계층적

*정희원, 한성대학교 컴퓨터공학부(교신저자)
접수일자: 2017년 9월 22일, 수정완료: 2017년 11월 5일
게재확정일자: 2017년 12월 8일

Received: 22 September, 2017 / Revised: 5 November, 2017 /
Accepted: 8 December, 2017

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

Dept. of Computer Engineering, Hansung University, Korea

구조를 기반으로 한 ORA-SS^[5], 시멘틱 네트워크 모델을 기반으로 한 CMXML^[6] 등이 있다.

또한, XML 문서를 개념적 모델로 변환하는 대표적인 연구결과로는 Xere^[7]와 Xer^[2]이 있다. 그러나 두 기법 모두 XML 스키마 문서를 개체 관계 다이어그램으로 간단한 알고리즘을 통해 자연스럽게 표현 가능한 장점은 있으나, 변환이 가역적이지 못한 결과로 추출된 개체 관계 다이어그램으로부터 원래의 XML 스키마 문서의 구조를 추론할 수 없는 단점을 내포하고 있다.

본 논문은 XML의 논리적 모델을 개념적 모델로 변환하는 역공학 기법을 제안한다. 변환절차는 XML 스키마 정의(XSD: XML Schema Definition)를 입력으로 받아, 변환을 위한 자료 구조를 생성하고, 변환규칙 알고리즘을 적용하여, 출력으로 CMXML 다이어그램을 생성한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 XML의 개념적 모델 도구인 CMXML에 대해 기술하고, 제 3장에서는 XML 스키마를 CMXML로 변환하기 위한 변환규칙을 정의한다. 제 4장에서는 XML 스키마 정보를 저장하기 위한 자료구조를 설정하고, 제 5장에서는 XML 스키마 문서에 제 3장에서 제시한 변환규칙을 적용하여 CMXML로 생성된 변환결과를 보여준다. 끝으로 제 6장에서 결론 및 향후 연구과제에 대해 기술한다.

II. CMXML

CMXML은 XML을 구성하는 각 개념들을 형식적으로 정의하고, 이 개념들을 심볼을 이용하여 다이어그램으로 표현하는 XML의 개념적 모델이다^[6].



















CMXML은 실제계의 객체를 표현하는 노드집합, 노드들의 관계를 표현하는 에지집합, 관계 타입을 표현하는 라벨집합, 노드와 에지에 적용되는 제약조건집합으로 구성된다. 표 1은 CMXML에서 사용하는 대표적인 다이어그램 심볼들을 보여준다.

노드는 비단말노드(non-terminal node)와 단말노드(terminal node)로 구분하며, 비단말노드는 다른 노드와 에지로 연결한다. 비단말노드는 노드명을 포함한 사각형으로 표현한다. 노드에 인스턴스가 존재하지 않고 단지, 타입만 존재하는 노드를 추상화노드(abstract node)라 하며, 추상화노드는 사각형 테두리를 점선으로 표현한다. 단말노드는 노드명과 함께 작은 사각형으로 표현하는데,

킷값은 작은 사각형을 색을 채우고, 다중값은 도메인 제약조건으로 표시하며, 리스트값은 작은 사각형을 이중으로 표시하고 내부를 채우고, 유니온값은 작은 사각형 안에 세로 바로 표시한다.

표 1. CMXML 다이어그램 기호

Table 1. CMXML diagram symbols

명칭	기호	적용
비단말노드		복합요소 표현
추상화노드		복합요소의 타입을 표현
단말노드		킷값, 원자값, 리스트값, 유니온값
콘텐츠 모델		all
		sequence
		choice
일반화		부분참여/중첩
		부분참여/분리
		전체참여/중첩
		전체참여/분리
연관화		비단말노드와 추상화노드 연결
		노드와 노드 연결
그룹		비단말노드 그룹
		단말노드 그룹
매핑대응수		one
		zero or one
		one or more
		zero or more

에지에는 라벨을 붙여 관계타입을 구분한다. 타입은 generalization, aggregation, association, typeof가 있다. 단일 예지는 부모노드에서 자식노드로 방향성 선분으로 표시하고, 에지그룹은 에지그룹에 속하는 모든 형제 노드들을 방향성 선분으로 묶어 하나의 선분으로 부모노드와 연결한다. 노드 및 에지에서의 준수해야 할 제약조건으로는 매핑대응수(mapping cardinality), 값의 범위(최소값, 최대값), 콘텐츠 모델(sequence, all, choice 등) 등이 있다.

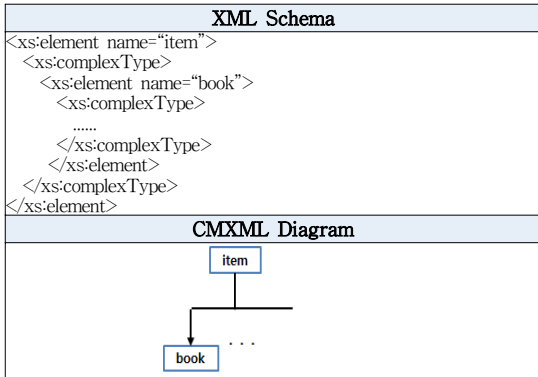
III. 변환규칙

이 장에서는 XML 스키마 문서를 CMXML 다이어그램으로 변환하기 위한 변환 규칙을 기술한다. 본 절에서

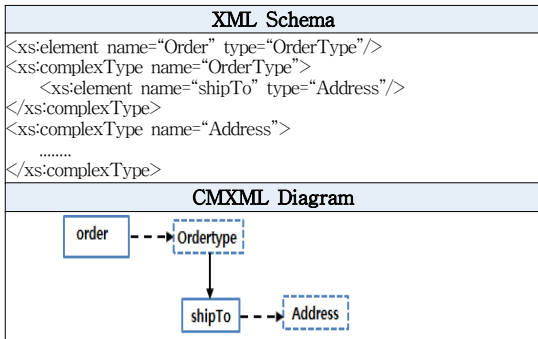
기술한 XML 스키마 문서는 완전한 XML 문법을 따르는 것이 아니고 단지 주요 관점만을 명세하고 있으며, CMXML 다이어그램 또한 변환의 논점이 되는 주요 부분만 기술한다.

1. 요소(element)

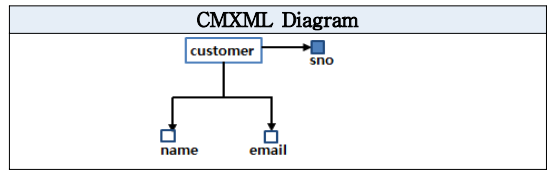
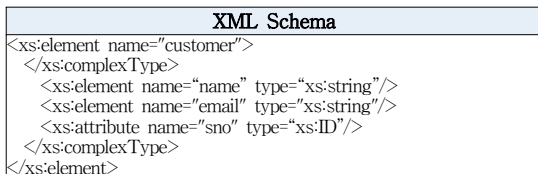
[규칙 1: 복합요소] 복합요소(complex element)는 비 단말노드로 표현하며 부모요소의 자식요소는 방향성 예지로 연결한다.



[규칙 2: 복합타입] 비 단말노드가 복합타입(complex type)으로 선언되어 있을 경우 복합타입은 추상화노드로 표현하며, 추상화노드명이 타입명이 된다.

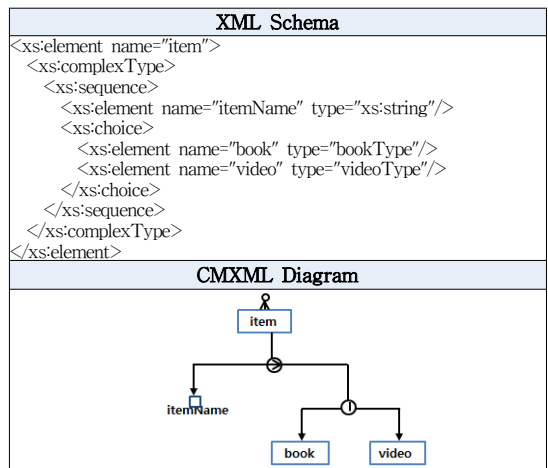


[규칙 3: 단순요소 및 속성] 단순요소(simple element) 및 속성(attribute)은 단말노드로 표현하며, 관련된 복합요소와 연결한다. CMXML에서는 단순요소와 속성을 구분하지 않고 모두 단말노드로 표현한다.



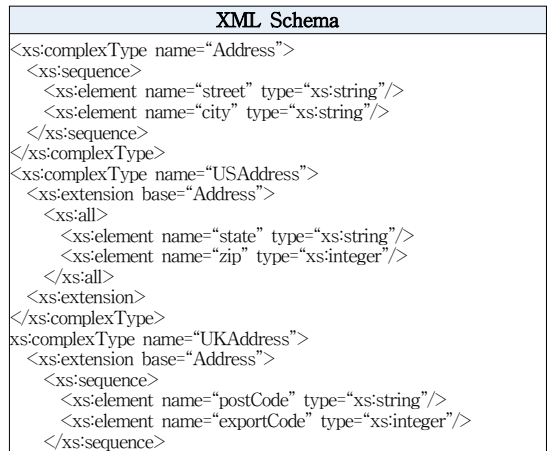
2. 콘텐츠 모델(content model)

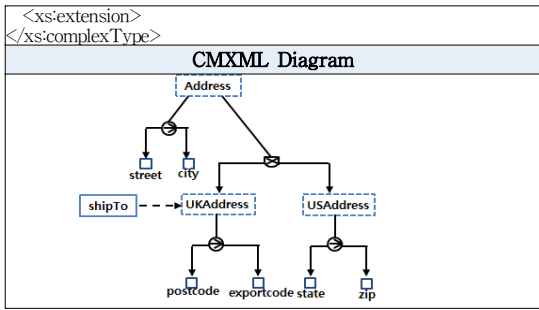
[규칙 4: 콘텐츠 모델] 콘텐츠 모델(content models)은 부모노드와 자식노드들 사이에 분기되는 지점에 원으로 표시하는데, 이 원이 없을 경우 묵시적으로 "all"을 나타내고, "sequence"일 경우 원 안에 화살표로 표시하고, "choice"일 경우 원 안에 세로바로 표시한다.



3. 유도요소(derived element)

[규칙 5: 유도 요소] 유도 요소(deriving element)는 기본 요소와 유도 요소 간에 일반화(generalization) 타입으로 표현한다. 즉, 기본 요소의 모든 자식 요소가 유도 요소의 자식 요소로 계승됨을 의미한다.

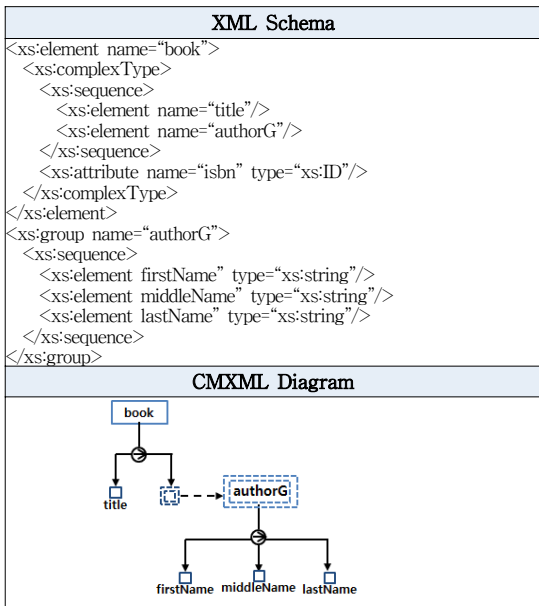




위의 다이어그램에서 복합타입인 Address 요소의 유도 요소로서 USAddress와 UKAddress가 선언되었음을 나타내고 있다. 복합요소 shipTo의 타입은 UKAddress이며, UKAddress는 자식요소로써 street, city, postcode, exportcode 임을 의미한다.

4. 그룹

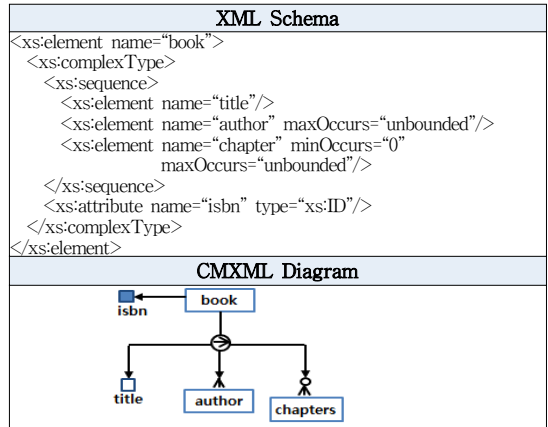
[규칙 6: 그룹] 그룹은 다수의 요소들을 하나의 단위로 묶어 재사용 가능한 콘텐츠를 사용하기 위해 정의한다. 그룹에는 요소그룹과 속성그룹이 있다.



위의 다이어그램에서 요소그룹 authorG는 book 요소 내의 자식요소에서 요소그룹으로 정의되어 있으며, 따라서 요소 book의 자식요소는 title, firstName, middleName, LastName이 된다.

5. occurs

[규칙 7: occurs] 부모요소와 자식요소 사이의 매핑 대응수는 크게 one, zero or more, one or more로 구분하며, “one”은 예지 끝에 아무 표시가 없으며, “1”, “zero”는 작은 원으로 표시하며, “more”는 까마귀 발로 표시하며 이들의 조합으로 매핑대응수를 나타낸다.



IV. 변환을 위한 자료구조

XML 스키마 문서를 분석하여 변환에 필요한 정보를 추출하기 위해 그림 1과 같이 XML 스키마 문서 정보를 저장하기 위한 6개의 자료구조를 정의한다. 이 중 그룹은 요소그룹과 속성그룹을 저장하기 위한 자료구조이고, Extension/Restriction은 자식요소가 부모요소 사이의 유도 관계 정보를 저장하기 위한 자료구조이다.

복합요소							
Name	Type	Min Occurs	Max Occurs	Content Model	Sub Element List	Attribute List	Next Element List
단순요소							
Name	Min Occurs	Max Occur	Type	Next Element List			
그룹							
Name	Group Type	Sub Element List	Attribute List				
콘텐츠 모델							
Type	Sub Content List	Element List	Next Element List				
Extension/Restriction Type							
Base	Name	Content Model	Sub Element List	Attribute List			
속성							
Name	Type	Next Attribute List					

그림 1. XML 스키마 정보 저장을 위한 자료구조
Fig. 1. Data Structure for XML Schema

V. 변환

본 장에서는 앞 장에서 기술한 변환규칙과 자료구조를 적용하여 XML 스키마 문서를 CMXML 다이어그램으로 변환하는 결과를 기술한다. 그림 2는 주문에 관한 XML 스키마 문서 예를 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:element name="Order">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="shipTo" type="Address"/>
      <xs:element name="customer">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="customerName" type="xs:string"/>
            <xs:element name="homeAddress" type="Address"/>
          </xs:sequence>
          <xs:attribute name="sno" type="xs:ID"/>
          <xs:attributeGroup ref="basicAttrGroup"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="item" type="Item"/>
    </xs:sequence>
    <xs:attribute name="orderno" type="xs:ID"/>
    <xs:attribute name="orderDate" type="xs:date"/>
  </xs:complexType>
</xs:element>
<xs:complexType name="Address">
  <xs:sequence>
    <xs:element name="street" type="xs:string"/>
    <xs:element name="city" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="USAddress">
  <complexContent>
    <xs:extension base="Address">
      <xs:sequence>
        <xs:element name="state" type="xs:string"/>
        <xs:element name="zip" type="xs:integer"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexType>
<xs:complexType name="UKAddress">
  <complexContent>
    <xs:extension base="Address">
      <xs:sequence>
        <xs:element name="postcode" type="xs:string"/>
        <xs:element name="exportCode" type="xs:integer"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexType>
<xs:complexType name="Item">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="quantity" type="xs:integer"/>
    <xs:choice>
      <xs:element name="book">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="authorGr" minOccurs="1" maxOccurs="unbounded"/>
            <xs:element name="title"/>
          </xs:sequence>
          <xs:attribute name="isbn" type="xs:ID"/>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
    
```

```

</xs:element>
<xs:element name="video">
  <xs:complexType>
    <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>
</xs:element>
</xs:choice>
<xs:attribute name="itemno" type="xs:ID"/>
</xs:complexType>
    
```

그림 2. 주문 XML 스키마 문서
 Fig 2. XML Schema Document for Order

그림 3은 그림 2의 XML 스키마 문서를 자료구조로 변환한 결과를 시각적으로 표현한 것이다. 그림 4는 그림 3의 자료구조 정보를 이용하여 생성한 CMXML 다이어그램을 보여준다.

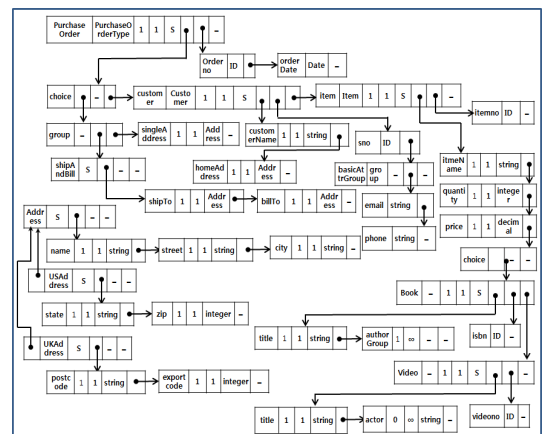


그림 3. 주문 XML 스키마 문서의 자료구조
 Fig. 3. Data Structure fro Order XML Schema Document

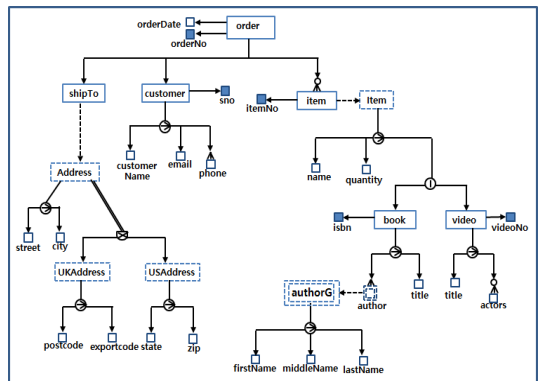


그림 4. 주문 CMXML 다이어그램
 Fig. 4. CMXML Diagram for Order

V. 결론

본 논문은 XML 스키마를 시각적으로 표현하기 위해 XML 스키마의 개념적 모델로 역변환하는 기법을 제안하였다. 이를 위해 XML 스키마의 개념적 모델인 CMXML을 제시하고, 이를 기반으로 XML 스키마를 구성하는 각 요소들을 CMXML 다이어그램으로 변환하기 위한 변환 규칙들을 제시하고, 이 변환 규칙을 적용하여 XML 스키마 문서를 확장된 CMXML 다이어그램으로 변환되는 결과를 보여주었다. 변환 절차는 ① XML 스키마 문서로부터 필요한 정보를 추출하여 XML 스키마 정보 자료구조에 저장하고, ② XML 스키마 정보 자료구조로부터 변환규칙과 변환 알고리즘을 통해 CMXML 다이어그램을 도출하였다.

향후 연구 과제로는 크게 본 연구의 완성도를 높이는 방향으로 진행될 예정이다. 본 연구에서 아직 해결하지 못한 부분으로 요소 타입에서 mixed, substitutionGroup, anyAttribute 개념 및 facet 개념을 지원할 수 있는 모델에 대해 연구가 진행할 예정이다.

References

- [1] World Wide Web Consortium, eXtensible Markup Language(XML) 1.0, <http://www.w3.org/TR/1998/REC-xml-19980210>, 1998.
- [2] A. Sengupta, S. Mohan, R. Doshi, "XER-Extensible Entity Relationship Modeling", Proceeding by deepX Ltd.
- [3] S. Jin, W. Kang, "Mapping Rules for ER to XML Using XML Schema" Proc. of the Southern Association for Information System Conference, 2007.
- [4] M. Necasky, I. Mlýnkova., "When Conceptual Model Meets Grammar: A Formal Approach to Semi-structured Data Modeling", Web Information Systems Engineering - WISE 2010, Lecture Notes in Computer Science, vol 6488. Springer, Berlin, Heidelberg.

DOI: https://doi.org/10.1007/978-3-642-17616-6_26.

- [5] G. Dobbie, W.Xiaoying, T.W. Ling, M.L. Lee, "ORA-SS: An Object-Relationship Attribute Model for Semi-Structured Data", Technicak Report, Dept. of Computer Science, National University of Singapore, 2000.
- [6] Y. Kim, "CMXML: A Conceptual Modeling Methodology for XML", The Journal of The Institute of Webcasting, Internet and Telecommunication, VOL. 15 NO. 4, 2015.
DOI: <https://doi.org/10.7236/JIIBC.2015.15.4.231>
- [7] 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.

저자 소개

김 영 응(정회원)



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

정 인 환(정회원)



- 2000년 KAIST 정보및통신공학과 박사
- 1985 ~ 1998년 삼성전자 시스템사업부 선임연구원
- 2001년 ~ 현재 한성대학교 컴퓨터공학과 교수
- <주관심분야 : 분산데이터베이스, 멀티미디어통신>

※ 본 연구는 한성대학교 교내연구비 지원 과제임.