http://dx.doi.org/10.7236/JIIBC.2013.13.2.149
JIIBC 2013-2-20

ERX : 개체 관계 모델로부터 XML 스키마 생성 도구

ERX : A Generation Tool of XML Schema based on Entity-Relationship Model

김영웅*

Young-Ung Kim

요 약 오늘날 대표적인 데이터베이스 설계 도구로 개체 관계 모델을 사용하고 있으며, 데이터를 표현하고 교환하는 표준 언어로 XML을 사용하고 있다. 그러나 많은 개체 관계 모델 제품들은 각각 서로 다른 표현형식을 사용하기 때문에 이들 제품들 사이에 호환성에 어려운 점이 있으며, XML은 언어가 갖는 복잡성으로 인해 XML을 이용하여 현실세계를 직접 설계하기에는 어려운 점이 있다. 본 논문은 이질적인 제품들 사이의 호환성을 제공하기 위해 개체 관계 모델을 XML 스키마로 변환하는 도구인 ERX(Entity-Relationship model to Xml)를 제안한다. ERX는 입력으로 개체 관계 다이어그램을 받아 이를 변환 규칙을 통해 XML 스키마를 출력한다. 변환 규칙에는 개체집합, 관계집합, 매핑 대응수(mapping cardinality), 일반화(generalization) 개념 등을 포함한다.

Abstract In these days, Entity-Relationship Model is the most popular modeling tool for designing databases, and XML is a de facto standard language for representing and exchanging data. But, because of many commercial products supporting Entity-Relationship Model use their's own representation formats, and thus it gives rise to difficulties the inter-operability between these products. In this paper, we propose an ERX, a generation tool of XML Schema from Entity-Relationship Model. ERX receives an Entity-Relationship Diagram as an input, transforms it based on transformation rules, and generates a XML Schema Definition as an output. Transformation rules contain entity set, relationship set, mapping cardinalities, and generalization.

Key Words : ERX, XML Schema Definition, Entity Relationship Model, Transformation Rule

I. 서 론

데이터베이스를 논리적으로 설계하는데 대표적으로 사용되는 개체 관계 모델(Entity-Relationship Model)은 기본적으로 개체와 개체 간의 관계를 이용하여 현실 세계의 데이터들을 표현하는 데이터 모델로써, 대표적인 제품으로는 Erwin[1], ER/Studio[2] 등이 있다.

한편, 웹상에서의 데이터 표현하고 교환하기 위한 사실상의 표준으로 사용되는 XML(eXtensible Markup Language)[3]은 데이터베이스 분야의 새로운 응용분야로 등장하였으며, 대표적인 응용으로는 데이터베이스 모델과 XML 사이의 스키마 변환을 들 수 있다.

개체 관계 모델은 개체와 그들 사이의 관계를 표현하는 데 중점을 두는 반면 XML 구조는 문서 위주의 표현

*정회원, 한성대학교 컴퓨터공학과
접수일자 2013년 2월 12일, 수정완료 2013년 3월 25일
게재확정일자 2013년 4월 12일

Received: 12 February 2013 / Revised: 25 March 2013 / Accepted: 12 April 2013
*Corresponding Author: yukim@hansung.ac.kr
Dept. of Computer Engineering, Hansung University, Korea

에 중점을 두고 있기 때문에 두 모델 사이에는 표현의 차이가 있어 직접적인 변환이 어렵다. 따라서 개체 관계 모델을 XML 스키마로 변환하기 위해서는 다음 사항들을 고려하여야 한다.

- 개체집합, 매핑 대응수에 따른 관계집합, 속성들을 어떻게 표현할 것인가.
- 주키, 외래키, 무결성 제약조건들을 어떻게 표현할 것인가.
- 개체 관계 모델이 갖는 평면 구조를 XML 스키마 구조 특성인 트리 구조로 어떻게 표현할 것인가.

본 논문은 이질적인 개체 관계 모델링 도구들 사이의 표준 환경을 제공하기 위해 개체 관계 모델을 XML 스키마 문서로 변환하는 도구인 ERX(Entity Relationship model to Xml)를 제안한다. 이를 위해 개체 관계 모델의 각 요소들을 XML 스키마로 변환하기 위한 변환규칙들을 제시하고, 변한규칙을 적용한 알고리즘을 기술한다. ERX는 입력으로 개체 관계 다이어그램을 받아 이를 변환 규칙을 통해 출력으로 XML 스키마 문서를 생성한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 개체 관계 모델을 XML로 변환하는 관련 연구를 기술하고, 제 3장에서는 개체 관계 모델의 주요 요소들을 XML 스키마로 생성하기 위한 변환규칙들을 기술한다. 제 4장에서는 제 3장에서 제안한 변환규칙을 근거로 변환 알고리즘과 자료구조를 제시하고, 사례를 통해 변환 결과를 보여준다. 끝으로 제 5장에서 결론 및 향후 연구과제에 대해 기술한다.

II. 관련 연구

데이터베이스와 XML 사이의 변환에 관한 연구로는 관계형 모델과 XML 사이의 변환에 관한 대표적 연구로 NeT(Nesting-based Translation) 및 CoT (Constraints-based Translation)[4]이 있고, XML 문서를 개체 관계 모델로 변환하는 대표적 연구로 Xere[5]가 있다. 본 연구와 직접 관련되는 연구로는 개체 관계 모델을 XML 문서로 변환하는 이론적인 연구와 개체 관계 모델링 도구들 사이에 호환성을 확보하기 위해 XML 문서로 변환하는 상업적인 연구가 있다. 전자의 대표적인 연구로는 ER-to-XML 기법이 있고 , 후자의 대표적인 연구로는 Erwin Data Modeler 제품에서의 연구가 있다.

ER-to-XML 기법[6]은 개체 관계 모델로부터 XML 스키마로 변환하기 위한 매핑 규칙과 알고리즘을 제안하고 있다. 매핑 규칙은 개체집합(strong/weak), 속성(simple/key/composite/multivalued/derived), 관계집합(unary/binary/n-ary)을 포함하고, 제안한 알고리즘은 매핑 규칙을 기반으로 개체 관계 다이어그램(Entity Relationship Diagram: ERD)을 입력받아 XML 스키마 문서(XML Schema Definition)를 출력하는 알고리즘이다.

이 기법은 매핑 규칙을 제안하고 있지만, 이 규칙은 개체 관계 모델이 가지는 요소들을 요소명과 키워드로 구분하는 극히 기본적인 규칙만 제시하고 있어, 간단하고 이해하기 쉬운 장점이 있는 반면, 개체 관계 모델이 가지는 특성인 외래키, 참조무결성, 매핑대응수에 대한 해결책을 제시하지 못하고 있다.

Erwin Data Modeler[1]는 상업적으로 널리 사용되는 대표적인 데이터베이스 모델링 도구로서, 기본적으로 논리적 데이터 모델링과 물리적 데이터 모델링을 지원한다. 또한 타 제품과의 호환성을 제공하기 위해 XML 형태의 export/import 기능을 제공한다. export/import 변환을 위해 메타데이터 인스턴스를 교환하는 방법을 제공하는 OMG(Object Management Group)의 CWM(Common Warehouse Metamodel)에 근거한 MIR(Meta Integration Repository) 메타모델을 기반으로 실현하였다[7].

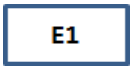
Erwin에서 제공하는 XML 변환 기법은 타 제품과의 export/import 기능에만 초점을 두고 있어 타 제품과의 호환성을 용이하게 해주는 장점이 있는 반면, 개체의 중복이 많이 발생하여 파일이 매우 커지는 단점이 있다. 실험에 의하면 약 백 개의 개체집합을 XML 스키마 문서로 변환하면 대략 10MB의 용량이 발생한다. 이런 이유로 xml 파일을 직접 분석하기가 매우 힘들며, xml 파일 내에 개체집합 정보가 중복으로 나타나기 때문에 xml 파일 수정시 비일관적인 상태를 초래할 수 있다.

III. 변환 규칙

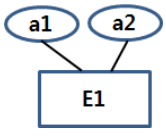
본 장에서는 개체 관계 모델을 구성하는 각각의 요소들을 XML 스키마로 변환하기 위한 변환 규칙을 기술하고, 이 규칙에 의해 생성되는 XSD를 기술한다. 본 논문

에서는 두 모델 사이의 변환 구조에 초점을 두고 있기 때문에 데이터 타입이나 도메인 무결성 제약조건 등 구조에 영향을 주지 않는 요소들에 대해서는 기술을 생략하였다.

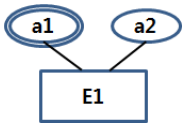
[규칙 1: 개체집합] 개체집합은 complexType을 갖는 요소(element)로 변환되며, 개체집합명(entity set name)이 요소명(element name)이 된다.

ERD	XSD
	<xs:element name="E1"> <xs:complexType> ----- </xs:complexType> </xs:element>

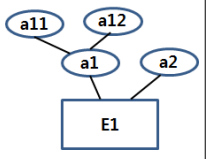
[규칙 2: 단순속성] 단순속성(simple attribute)은 그 속성이 지정된 element의 attribute로 생성한다. 속성의 순서는 무관하므로 <all> 태그를 적용한다.

ERD	XSD
	<xs:element name="E1"> <xs:complexType> <xs:all> <xs:attribute name="a1"/> <xs:attribute name="a2"/> <xs:all> </xs:complexType> </xs:element>

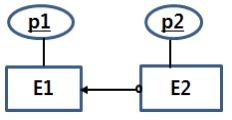
[규칙 3: 다중값 속성] 다중값 속성(multi-valued attribute)은 그 속성이 지정된 element의 자식 element로 생성하고, minOccurs와 maxOccurs를 적용한다.

ERD	XSD
	<xs:element name="E1"> <xs:complexType> <xs:element name="a1" minOccurs="1" maxOccurs="unbounded"/> <xs:attribute name="a2"/> </xs:complexType> </xs:element>

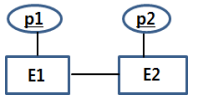
[규칙 4: 복합속성] 복합속성(composite attribute)은 attributeGroup으로 선언하고 그 속성이 지정된 element에서 이 attributeGroup을 ref로 참조한다.

ERD	XSD
	<xs:element name="E1"> <xs:complexType> <xs:attributeGroup ref="a1"> <xs:attribute name="a2"/> </xs:complexType> </xs:element> <xs:attributeGroup id="a1"> <xs:attribute name="a11"/> <xs:attribute name="a12"/> </xs:attributeGroup>

[규칙 5: 1:n 관계집합] 1:n 관계집합은 관계에 참여하는 두 개체집합을 element로 정의하고, 1측 element에는 참여도(participation)가 부분참여(partial)일 경우 minOccurs ="0"으로 정의하고 n측의 element에는 maxOccurs= "unbounded"로, 부분참여일 경우 minOccurs="0", 전체참여(total)일 경우 minOccurs="1"로 정의하고 1측의 element를 ref로 참조한다.

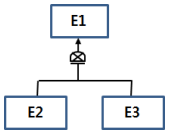
ERD	XSD
	<xs:element name="E1" minOccurs="0"> <xs:attribute name="p1" type="xs:ID"/> </xs:element> <xs:element name="E2" minOccurs="0" maxOccurs="unbounded"> <xs:element ref="E1"/> <xs:attribute name="p2" type="xs:ID"/> </xs:element>

[규칙 6: m:n 관계집합] m:n 관계집합의 경우 관계에 참여하는 두 개체집합을 element로 정의하고, 두 개의 element를 동시에 참조하기 위해 새로운 element를 생성하여 그 밑에 자식 element를 두어 두 개체집합을 참조하게 하고, minOccurs와 maxOccurs를 적용한다.

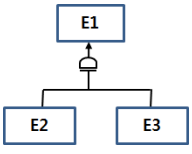
ERD	XSD
	<xs:element name="E1"> <xs:attribute name="p1" type="xs:ID"/> </xs:element> <xs:element name="E2"> <xs:attribute name="p2" type="xs:ID"/> </xs:element> <xs:element name="E1/E2"> minOccurs="0"

	maxOccurs="unbounded"> <xs:element ref="E1"/> <xs:element ref="E2"/> </xs:element> </xs:element>

[규칙 7: 분리 제약조건 일반화] 분리 제약조건 일반화(generalization with disjoint constraints)는 하나의 상위 개체는 단지 하나의 하위 개체집합에만 속해야하는 제약조건이다. 일반화 관계에 참여하는 각각의 개체집합을 element로 선언하고, 상위 개체집합 element에서 하위 개체집합 element를 choice로 참조하고, 하위 개체집합의 element에는 상위 개체집합 element를 extension으로 정의한다.

ERD	XSD
	<xs:element name="E1"> <xs:choice> <xs:element ref="E2"/> <xs:element ref="E3"/> </xs:choice> </xs:element> <xs:element name="E2"type="t2"/> <xs:complexType name="t2"> <xs:extension base="E1"/> </xs:complexType> <xs:element name="E3"type="t3"/> <xs:complexType name="t3"> <xs:extension base="E1"/> </xs:complexType>

[규칙 8: 중첩 제약조건의 일반화] 중첩 제약조건의 with overlapping constraints)는 하나의 개체는 다수의 하위개체집합에 속할 수 있는 제약조건이다. 중첩 제약조건의 일반화와 분리 제약조건의 일반화의 유일한 차이점은 하위 개체집합을 참조할 때 choice 선언이 없다.

ERD	XSD
	<xs:element name="E1"> <xs:element ref="E2"/> <xs:element ref="E3"/> </xs:element> <xs:element name="E2"type="t2"/> <xs:complexType name="t2"> <xs:extension base="E1"/> </xs:complexType> <xs:element name="E3"type="t3"/> <xs:complexType name="t3"> <xs:extension base="E1"/> </xs:complexType>

IV. ERX : ERD로부터 XSD 생성 도구

그림 1은 개체 관계 모델로부터 XML 스키마 문서를 생성하는 과정을 보여준다.

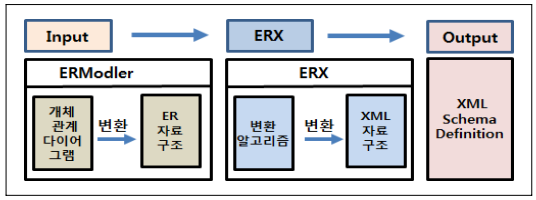


그림 1. XML 스키마 생성 과정
Fig. 1. Process of XML schema generation

개체 관계 모델로부터 XML 스키마 문서를 생성하는 과정은 ERModeler를 이용하여 개체 관계 다이어그램을 작성하면 이를 내부적인 자료구조로 변환하고, 이를 입력으로 제 III장에서 기술한 변환 규칙을 적용한 알고리즘을 통해 XML 스키마 정보를 추출하여 XML 자료구조로 변환하고, 이를 토대로 XSD를 생성하는 과정을 거친다.

ER_Modeler[8]는 ERD를 윈도우 상에서 작성할 수 있는 ERD 편집도구 기능과 생성된 ERD를 데이터베이스 테이블로 자동 정의해주는 DDL 생성 기능을 제공해주는 데이터베이스 설계도구이다.

1. ERX

ERX는 앞 절에서 기술한 개체 관계 모델의 자료구조를 입력으로 하여 변환 알고리즘을 적용하여 XML 스키마 문서를 생성하기 위한 자료구조를 생성한다. 변환 알고리즘은 그림 2와 같다.

Input : ERModeler에 의해 생성된 자료구조
Output : XML 스키마 문서
1. 루트 element를 생성한다. 이때 element name은 ERD 파일명으로 선언한다.
2. Entity set List로부터 모든 개체집합을 읽어 3에서 6을 반복한다.
3. 전역 개체집합(외래키를 가지지 않는 개체집합)을 루트 element의 child-element로 생성한다. [규칙 1]
4. 단순 속성은 해당 element의 attribute로 생성한다. 속성이 주키일 경우 ID type을 생성하고, not null일 경우 use="required"를 설정한다. [규칙 2]
5. 다중값 속성은 해당 element의 sub-element로 생성하고, minOccurs와 maxOccurs를 적용한다. [규칙 3]
6. 복합 속성은 attributeGroup으로 선언하고 해당 element 내에서 이 attributeGroup을 ref로 참조한다. [규칙 4]

7. Entity set List로부터 모든 관계집합을 읽어 8에서 11을 반복한다.
8. 대응수가 1:n 관계일 경우 참여도에 따라 각각의 element에 minOccurs, maxOccurs 값을 결정하고 n측의 element에서 1측의 element를 ref로 참조한다. 이 경우 두 element는 동일한 parent-element를 갖도록 설정한다. [규칙 5]
9. 대응수가 m:n 관계일 경우 새로운 element를 생성하여 두 element를 ref로 참조한다. 이 때 새로 생성된 element 명은 element1명/element2명으로 생성하고, 두 element와 동일한 parent-element를 갖도록 설정한다. [규칙 6]
10. 관계집합이 분리 제약조건의 일반화 관계일 경우 상위 element에서 각각의 하위 element를 choice로 참조하고, 각각의 하위 element에는 상위 element를 extension으로 설정한다. [규칙 7]
11. 관계집합이 중첩 제약조건의 일반화 관계일 경우 상위 element에서 각각의 하위 element를 ref로 참조하고, 각각의 하위 element에는 상위 element를 extension으로 설정한다. [규칙 8]

그림 2. XML 스키마 생성 알고리즘
Fig. 2. Algorithm for XML schema generation

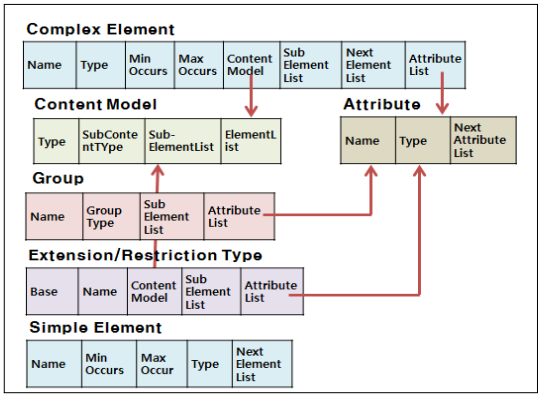


그림 3. XML 스키마 자료구조도
Fig. 3. Data structures of XML schema

그림 3은 그림 2의 알고리즘을 거쳐 생성하는 XML 스키마 문서 정보를 저장하기 위한 자료구조를 보여준다. 그림 3에서 Complex Element, Simple Element는 각각 복합요소와 단순요소 정보를 저장하기 위한 자료구조이고, Attribute는 속성을 저장하기 위한 자료구조이며, Group은 요소그룹과 속성그룹을 저장하기 위한 자료구조이고, Content Model은 자식요소가 바로 컨텐츠 모델로 정의되었을 때의 정보를 저장하기 위한 자료구조이며, Extension/Restriction은 자식요소가 부모요소와 확장/축소 관계로 정의될 때의 정보를 저장하기 위한 자료구조이다.

2. 사례

본 절에서는 앞에서 기술한 변환 알고리즘과 자료구조를 토대로 ER_Modeler에서 생성한 ERD를 XML 스키마로 변환한 예를 보여준다. 그림 4는 ER_Modeler에서 작성한 은행 ERD의 샘플을 보여주며, 그림 5는 그림 4의 ERD를 변환 알고리즘을 적용한 후에 생성되는 XML 스키마 문서 정보의 자료구조를 표현한 것이다.

그림 6은 그림 5의 자료구조를 통해 생성된 최종 XML 스키마 문서를 보여준다. 그림 6의 XML 스키마 문서는 XML 스키마 변환 구조에 영향을 주지 않는 요소들은 생성과정에서 생략하였다. ER_Modeler에서는 다중 값 속성과 복합 속성은 각각을 개체집합으로 표현하고 대응하는 개체집합과 1:n 관계로 연결해서 해결하고, m:n 관계집합은 두 개의 1:n 관계집합으로 풀어서 표현한다.

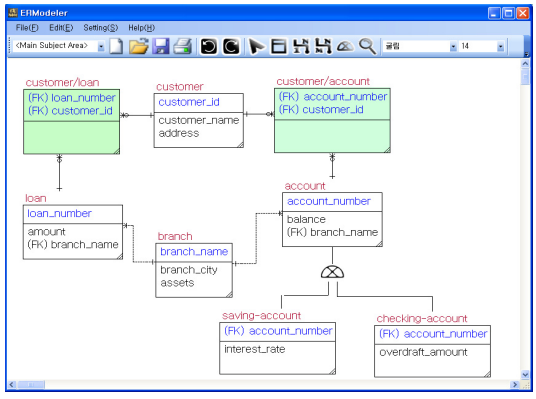


그림 4. 은행 개체 관계 다이어그램
Fig. 4. Entity-Relationship diagram for bank

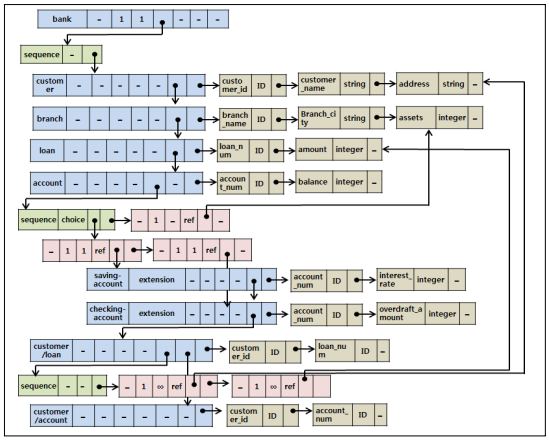


그림 5. 은행 XML schema 자료구조
Fig. 5. Data structure for bank of XML schema

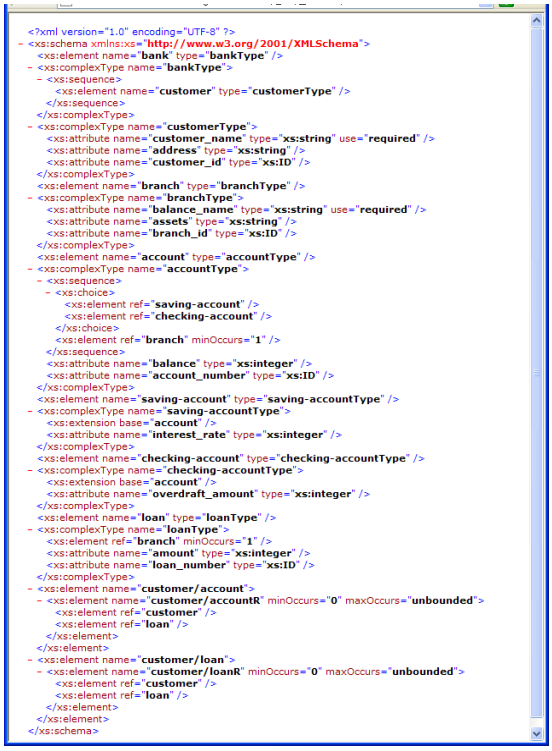


그림 7. 은행 ERD로부터 XSD 변환결과
Fig. 7. Result of XSD from Fig. 6

V. 결 론

본 논문은 이질적인 개체 관계 모델링 도구들 사이의 변환을 위한 표준 환경을 제공하기 위해 개체 관계 모델을 XML 스키마 문서로 변환하는 도구인 ERX를 제안하였다. 이를 위해 개체 관계 모델의 각 요소들을 XML 스키마로 변환하기 위한 여덟 가지 변환 규칙들을 제시하고, 제안한 변환규칙을 토대로 변환 알고리즘도 기술하였다. 또한 사례를 통해 개체 관계 모델로부터 XML 스키마 정보 저장을 위한 자료구조를 생성하고, XML 스키마 문서를 생성하는 과정도 기술하였다.

향후 연구 과제로는 크게 본 연구의 완성도를 높이는 연구와 역으로 XML 스키마 문서로부터 개체 관계 모델을 생성하는 기법 연구로 진행될 예정이다. 본 연구의 완성도를 높이기 위해서는 데이터베이스에서 제공하는 데이터 타입을 XML 스키마에서 제공하는 데이터 타입과의 변환, 다중 상속(multiple inheritance)의 지원, element의 중복해결을 지원할 수 있도록 연구가 진행될 계획이며, 또한 XML 스키마를 개체 관계 모델로 역변환이 가능하도록 기존의 개체 관계 모델을 변경, 확장하는 방법에 대한 연구를 진행할 예정이다.

참고 문헌

[1] http://www.ca.com/events/webcasts/na/Understanding-CA-ERwin-Design-Layer-Architecture/53826.aspx

[2] http://www.embarcadero.com/products/er-studio-data-architect

[3] World Wide Web Consortium, eXtensible Markup Language(XML) 1.0, http://www.w3.org/TR/1998/REC-xml-19980210, 1998.

[4] D. Lee, M. Mani, F. Chit, and W. Chu, “NeT&CoT: Translating Relation Schemas to XML Schemas using Semantic Constraints”, 11th ACM Int’l Conference on Information and Knowledge Management 2002.

[5] 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.

[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] http://www.metaintegration.net/Products/MIMB/Specifications/

[8] I. H. Jung, Y. U. Kim, “ ER-Modeler: A logical Database Design Tool based on Entity-Relationship Model“, Journal of The Institute of Webcasting Internet and Telecommunication′, vol11, no.8, pp11-17, Oct 2011.

※ 본 연구는 한성대학교 교내학술연구비 지원과제 임

저자 소개

김 영 웅(정회원)

- 1993년 : KAIST 전산학과 박사
- 1984년 ~ 1997년 : KT 통신망연구소
- 1997년 ~ 현재 : 한성대학교 컴퓨터 공학과 교수

<주관심분야 : 소프트웨어 신뢰도, 소프트웨어 설계, 데이터 모델링>