

# UML 클래스 다이어그램을 이용한 XML-GL 질의 모델링

최 봉 진<sup>†</sup> · 유 춘 식<sup>\*\*</sup> · 김 용 성<sup>\*\*\*</sup>

## 요 약

최근 다양한 분야에서 폭넓게 활용되고 있는 XML 문서는 유연하고도 개방적인 특성으로 인해 정보교환이나 전송을 위한 수단으로 널리 이용되고 있다. 한편 XML 문서를 위한 시각적, 직관적 질의 언어인 XML-GL은 질의에 대한 의미와 결과 문서의 구조를 시각적으로 표현할 수 있기 때문에 XML 문서에 대한 구조 검색과 정보의 공유가 용이하다. 그리고 UML은 정해진 표기법과 다양한 다이어그램을 이용하여 객체지향 분석과 설계를 위한 도구로 사용되고 있다. 본 논문은 UML 클래스 다이어그램을 이용한 XML-GL 질의 모델링 방안을 제안한다. 또한, 모델 구성요소에 대한 제약사항을 정확하게 묘사하기 위하여 객체제약언어(Object Constraint Language)로 정의하였다. 이를 통해 XML 문서를 객체지향 데이터로 변환하여 저장/관리할 수 있으며, UML 클래스 다이어그램을 이용한 질의 모델링 방법을 적용하여 보다 효율적으로 XML 문서를 검색할 수 있다.

키워드 : XML, XML-GL, XML-GDM, UML, 시각 질의 모델링, XML 스키마

## XML-GL Query Modelling using UML Class Diagram

Bong-Jin Choi<sup>†</sup> · Chun-Sik Yoo<sup>\*\*</sup> · Yong-Sung Kim<sup>\*\*\*</sup>

## ABSTRACT

Nowadays, XML has been favored by many companies internally and externally as a means of sharing and distributing data, due to its open-architectural structure. XML-GL, a graphical query language for document has the advantage of containing both structuring and defining of itself. By incorporating UML an XML document can become object-oriented and can be represented by graphical means. This paper proposes a XML-GL query modeling solution by using UML class diagrams. In order for the modeled objects to be properly restricted, the Object Constraint Language has been defined. This process converts XML documents into Object-Oriented data and combined with UML class diagrams, searches for XML documents can be increased.

Key Words : XML, XML-GL, XML-GDM, UML, Visual Query Modelling, XML Schema

## 1. 서 론

최근 급증하는 인터넷의 활용으로 웹 기반의 멀티미디어 문서를 보다 용이하게 저장, 관리하기 위한 방법으로 XML 문서를 객체지향적인 접근법에 의해 모델링하고 저장하기 위한 많은 연구나 시스템들이 있다[9]. 이러한 XML 문서에 대한 객체지향적 모델링을 위한 대표적인 도구가 UML(Unified Modeling Language)[10]이다. UML은 초기에는 소프트웨어 개발을 위한 통합된 방법론으로 활용되었으나 현재는 다양한 객체에 대한 모델링 언어로 보다 많이 활용되고 있다. 즉 현재 UML은 객체지향 분석과 설계를 위하여 클래스 다이어그램을 비롯한 다양한 다이어그램들을 지원하

며, 이들로부터 각종 데이터베이스 스키마와 객체지향 코드를 생성해 주는 도구로 널리 이용되고 있다.

본 논문에서는 웹 문서의 표준으로 널리 활용되는 XML 문서에 대한 사용자의 활용도를 높이고 XML 문서에 대한 질의를 효율적으로 지원할 수 있는 질의 모델링 방안을 제안한다. 이를 위하여 XML 문서, XML DTD, XML의 질의 등을 UML의 클래스 다이어그램을 이용한 XML-GL 질의를 모델링하여 객체지향 모델의 다양한 속성을 갖는 XML 문서에 대한 질의를 생성한다. 따라서 XML 문서를 UML 기반의 객체지향 그래픽 데이터 모델로 시각화하여 변환, 저장, 관리함으로써 사용자가 XML 문서에 대한 검색 및 질의를 직관적이고 시각적으로 표현할 수 있다. 따라서 본 논문의 공헌도는 다음과 같다. 첫째, 기존의 XML 기반의 질의 언어와 비교하면 다양한 객체지향적 특징을 가지고 있으며 객체 모델링 도구로 널리 활용되고 있는 UML의 표기법을 이용하게 되어 새로운 질의 표현 방식을 익히지 않더라도

<sup>†</sup> 중신회원 : EAI, BPM Consultant

<sup>\*\*</sup> 중신회원 : 전북대학교 정보전산원 학사행정지원팀

<sup>\*\*\*</sup> 정 회 원 : 전북대학교 정보전산원 원장

논문접수 : 2006년 8월 1일, 심사완료 : 2007년 1월 8일

XML 기반의 웹 문서에 대한 시각적이고 직관적인 질의를 수행할 수 있다. 둘째, 제안된 방식은 이를 통해 XML 문서의 내용, 질의의 문법과 의미를 UML 클래스 다이어그램이라는 동일한 모델링 도구를 적용함으로써 XML 문서를 객체지향 데이터베이스에 저장, 검색하는 등 일련의 모든 과정을 일관적으로 수행할 수 있게 되는 장점이 있다.

### 2. 관련 연구

XML 데이터를 대상으로 하는 그래픽 질의 언어에는 XML-GL[2]과 BBQ[13]가 있다. 웹 문서의 특정 상단일 문서에 대한 질의 보다 복수 문서에 대한 질의의 필요성이 더 부각되고 있는 실정에서 복잡한 문서 구조를 보다 쉽게 접근할 수 있는 시각화 방법으로 XML-GL과 같은 그래픽 사용자 인터페이스가 제안되었다. 그래픽 질의 언어를 대표하는 예인 XML-GL은 XML 그래프에 의해 XML 문서와 DTD를 그래프로 표현하는 방법으로, XML-GL은 그래프 기반의 논리적 언어로 전통적인 그래픽 데이터 모델들과 질의 언어들인 Graph log, Good, G-log 같이 그래프를 패턴으

로 취급하고, 질의 결과를 찾기 위해서 그래프 기반으로 매칭시키는 방식에 의해서 처리된다. XML-GL에 직접적으로 영향을 준 언어는 G-log로 논리적인 반응을 그래픽 언어로 복잡한 개체질의 표현이 용이하며 훗날 WG-log로 발전되어 웹기반의 질의 언어로 활용되고 있다.

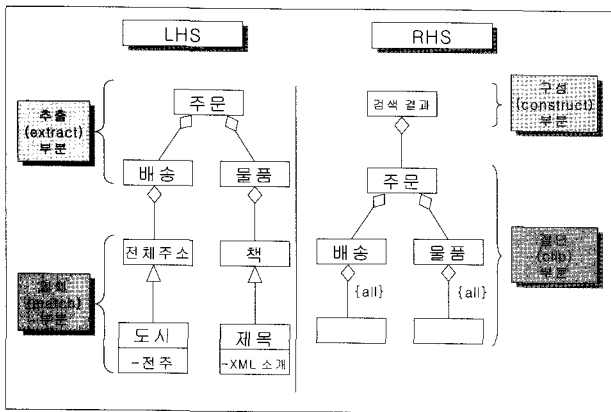
본 논문에서는 XML-GL을 객체지향 데이터베이스에 효과적으로 저장하여 재사용하기 위해 XML-GL을 UML 다이어그램 집합으로 변환하는 규칙을 정의하고, 이를 기반으로 객체지향적, 시각적으로 표현된 그래픽 질의를 통하여 XML-GL의 데이터 모델인 XML-GDM으로 표현된 XML 문서를 검색하는 시스템을 제안한다.

### 3. XML-GL

XML-GL은 XML-GDM에 의해 표현되고 DB에 저장된 XML 문서를 검색하고 검색 결과를 재구성하기 위한 그래픽 질의 언어이다. XML-GL의 질의 구문은 사용자가 원하는 정보를 표현하는 LHS(Left-Hand Side) 부분, 검색된 질의 결과의 구조를 표현하는 RHS(Right-Hand Side) 부분

〈표 1〉 XML-GL 표기법

Feature		Graphic Representation	Feature		Graphic Representation	
Node	List		Arc	Star		
	Index			Binding Edge		
	Computation				ANY	
Arc	Group_By		Label	unary (예) 수량>9		
	Aggregation (SUM, MIN, MAX, COUNT)			binary (예) 수량A > 수량B		
	Arithmetic	Compute-withproperty			Order(ASC, DESC)	
Compute-withvalue			Negation			



(그림 1) 질의 구성 요소

그리고 LHS와 RHS의 엘리먼트들을 서로 바인딩하는 요소로 구성된다. 즉 LHS 부분은 XML 입력문서들에 대해 매칭(matching)되는 질의 패턴을 명시하는 부분이고, RHS 부분은 원하는 출력 내용을 정의하기 위해서 사용된다. 이 두 부분 모두가 XML 질의 그래프(확장된 XML-GDM)에 기반을 두고 표현된다. LHS는 다시 추출 부분(extract part)과 일치 부분(match part)으로 구성되며, RHS는 절단 부분(clip part)과 구성 부분(construct part)으로 구성된다.

이에 대한 개념도는 (그림 1)과 같으며, 각 구성 요소에 대한 설명은 다음과 같다.

추출(Extract) 부분은 질의가 수행될 대상 문서와 대상 엘리먼트를 명시한다. SQL과 비교해보면, from 절에 해당하는 부분이며, 일치(Match) 부분은 추출 부분에 명시된 대상 엘리먼트들에 대하여 적용될 논리적인 조건을 규정하는 부분이다. SQL의 where 절에 해당한다. 그리고 절단(Clip) 부분은 일치 부분의 조건을 만족하는 엘리먼트들의 하위 엘리먼트를 규정한다. 즉 검색 결과로 제시될 문서에 포함될 엘리먼트들의 구조를 지정하는 부분이다. SQL의 select 절에 해당한다. 마지막으로 구성(Construct) 부분은 검색 결과문에 포함될 새로운 엘리먼트들의 구조를 명시하고 이들 엘

리먼트들과 LHS에 의해 추출된 엘리먼트들 사이의 관계를 규정한다. SQL과 비교해보면, 사용자가 질의 결과로부터 새로운 튜플을 설계할 수 있도록 하는 create view 문의 확장이라고 볼 수 있다.

XML-GL의 표기법은 <표 1>과 같으며 XML-GDM의 표기법을 확장하여 사용한다.

### 3.1 질의 유형

질의 구성방법에 따른 XML 문서에 대한 단순질의의 유형은 추출-절단 질의, 추출-일치-절단 질의, 추출-일치-절단-구성 질의로 구분할 수 있다. 먼저 추출-절단 질의는 XML 질의의 가장 단순한 형태로, XML 문서의 일부분을 추출하고 추출된 데이터를 포함하는 새로운 문서를 출력한다. 추출-일치-절단 질의에서 일치 부분은 선택된 술어들의 논리적 조건을 규정하며 질의의 왼쪽 그래프를 확장하며, 질의의 조건은 추출 부분에서 여러 목적 엘리먼트들 뿐만 아니라 하위 엘리먼트들까지도 포함한다. 그리고 추출-일치-구성-절단 질의는 여러 엘리먼트들의 하위 엘리먼트들의 조합 또는 대상 문서들로부터 추출된 엘리먼트들로 구성된 새로운 엘리먼트를 생성할 때 사용한다.

### 3.2 XML-GL의 복합질의 처리

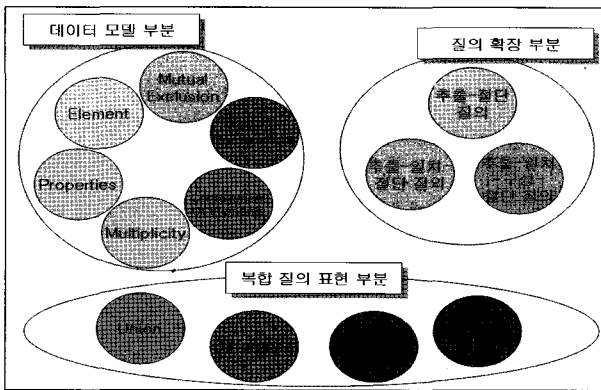
LHS나 RHS에 하나의 XML 질의 그래프만을 갖는 단순 질의와는 달리, LHS나 RHS에 두 개 이상의 XML 질의의 그래프를 갖는 질의를 복합질의(complex query)라고 하며, XML-GL은 다른 XML 질의 언어에 비해 복합질의를 직관적으로 처리할 수 있다. <표 2>는 XML-GL에서 Union, Difference, Cartesian Product, Heterogeneous Union과 같은 복합질의를 처리하는 예를 나타내고 있다.

## 4. UML 클래스 다이어그램으로의 사상하기 위한 XML 질의 모델링

본 장에서는 XML-GDM에 의해 저장된 XML 데이터,

<표 2> XML-GL의 복합질의 처리의 예

Query Operations	Graphic Representation	Query Operations	Graphic Representation
Union		Cartesian Product	
Difference		Heterogeneous Union	



(그림 2) 모델링 구성 요소

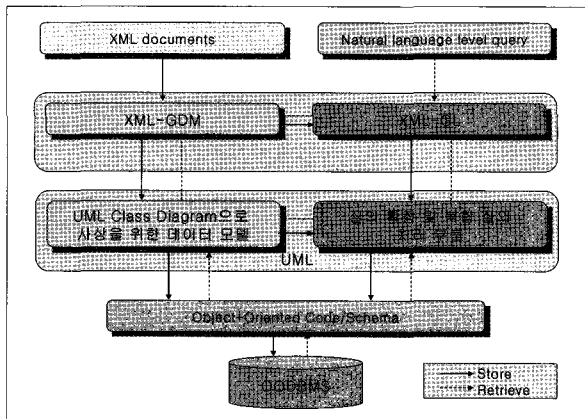
XML 문서, 비구조적 문서와 질의를 OODB에 저장하고 이를 검색 및 재구성할 수 있도록 UML 클래스 다이어그램으로 변환하는 방안을 제안한다.

UML 다이어그램으로 사상하기 위한 XML 문서와 질의 모델링은 XML-GL에 기반하며 (그림 2)와 같이 XML-GDM을 UML로 변환하는 데이터 모델 부분, XML-GL의 질의 유형에 따라 UML로 변환하는 질의확장 부분, 그리고 XML-GL의 복합질의 변환한 복합질의 표현 부분으로 나눌 수 있다. 본 논문에서는 질의확장 부분과 복합질의 표현 부분만을 대상으로 하며, 데이터 모델 부분에 대한 사상은 별도로 다루기로 한다.

4.1 UML 클래스 다이어그램으로의 사상을 위한 데이터 모델

XML-GDM이 XML-GL을 이용하여 XML 문서를 검색할 수 있도록 데이터베이스에 저장하기 위한 데이터 모델이라면, UML 클래스 다이어그램으로 사상을 위한 데이터 모델은 이를 UML로 변환하여 OODB에 저장할 수 있게 해준다.

(그림 3)은 XML 문서를 구조화된 XML-GDM과 UML 기반의 그래픽 데이터 모델을 통해 OODBMS에 저장하고, 자연어 수준의 질의가 들어왔을 때 XML-GL과 질의확장 및 복합질의 처리 부분을 이용하여 검색한 결과를 출력해주는 전체 시스템 구성도이다.



(그림 3) UML 클래스 다이어그램으로 사상을 위한 데이터 모델과 전체 시스템 구성도

4.1.1 질의 구성요소

본 논문에서의 모델링 대상이 XML-GL로 표현된 XML 문서 질의이므로 UML 클래스 다이어그램으로 사상된 XML 질의를 구성요소도 XML-GL의 질의 구성방법과 동일하다. 즉 질의는 크게 LHS와 RHS 부분으로 구성된다. LHS는 다시 추출 부분과 일치 부분으로 구성되며, RHS는 절단 부분과 구성 부분으로 구성된다.

4.1.2 XML-GDM 사상 규칙

UML은 많은 모델링 요소들을 제공하고 있지만 XML 문서를 UML 클래스 다이어그램으로 사상하기 위한 XML 문서 객체 모델링을 위해서는 보다 명확하게 생성 규칙을 설정해야 한다. 이를 위해 본 논문에서는 XML-GDM으로 표현된 XML 문서의 DTD와 XML 문서 인스턴스로부터 UML 클래스 다이어그램을 생성하기 위해 다음과 같이 정의한다.

[정의 1] XML-GDM의 요소들은 UML 클래스 다이어그램 집합으로 사상된다.

먼저 XML-GDM으로 표현된 XML 문서를 UML 클래스 다이어그램으로 사상하기 위한 방법을 요약하면 다음 <표 3>과 같다.

<표 3> XML-GDM과 UML 사상

	XML-GDM	UML 기반 그래픽 데이터 모델
엘리먼트	Default	Class
	Containment of Elements	집단화 관계
	Order of Sub-element	{ordered} 제한조건
	Mutual Exclusion	{XOR} 제한 조건
값 (Property)	Element with PCDATA Content	"String" Class로부터 상속
	Mixed Content	{XOR} 제한 조건
	ID Attributes	Private Attribute
	IDREF Attributes	Public Attribute
다중성	0:1	'0..1'
	1:N	'1..*'
	0:N	'0..*'

4.2 XML-GL 질의 사상 규칙

UML내에서 많은 모델링 요소들을 제공하고 있지만 XML-GL의 질의를 모델링하기 위해서는 보다 명확한 생성 규칙을 설정해야 한다. 이를 위해 본 연구에서는 XML-GL 기반으로 XML 문서와 질의의 인스턴스로부터 UML 클래스 다이어그램을 생성하기 위해 다음과 같은 정의와 규칙들을 기술한다.

**[정의 1]** XML-GL의 요소들을 UML 클래스 다이어그램의 집합으로 사상한다.

4.2.1 UML 클래스 다이어그램을 이용한 XML-GL 단순 질의 사상

위의 생성규칙에 의해 생성된 UML 클래스 다이어그램을 이용하여 기존의 XML-GDM 데이터와 XML 문서들을 OODB 에 저장한 후 이를 검색하거나 재구성 하려면 XML-GL 질의를 UML로 사상하여야 한다. XML-GL의 질의 유형에 따라 추출-절단 질의, 추출-일치-절단 질의, 추출-일치-구성-절단 질의로 구별하고 이에 알맞은 UML 표기법을 찾아 UML로 사상시킨다.

<표 4> XML-GL과 UML 사상

XML-GL		UML	
추출-절단 질의	Star	{all} 제약조건	
	ANY	빈 클래스, 집단화 관계	
	Binding Edge	<<bind>> stereotype, 의존관계	
	Order(ASC, DESC)	{ASC DESC} 제약조건 정렬의 기준 상속	
추출-일치-절단 질의	Arithmetic	Compute-with-property	클래스 Properties로부터 상속
		Compute-with-value	클래스 값으로부터 상속
	Predicate	unary	private attribute
		binary	private attribute, (속성 값) 제약조건
Negation	{~} 제약조건		
추출-일치-구성-절단 질의	List	'1..*' Multiplicity	
	Index	Group_By	'1..*' Multiplicity, {Group_By} 제약조건
		Aggregation (SUM, MIN, MAX, COUNT)	'1..*' Multiplicity and {Aggregation 종류} 제약조건

(1) 추출-절단 질의

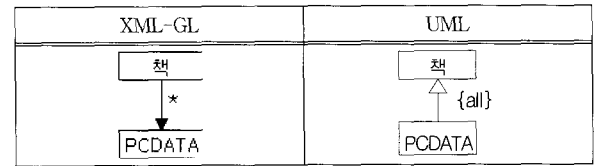
추출-절단 질의의 왼쪽 부분은 질의의 추출부분을 나타낸다. (그림 5)에서 추출부분은 하나의 엘리먼트만 추출하지만 일반적으로는 여러 개의 엘리먼트를 포함할 수 있다. 또한 추출부분은 (그림 5)와 같이 URL을 포함할 수 있다. 추출-절단 질의의 오른쪽 부분은 질의의 절단부분을 나타내며 결과 문서의 DTD를 UML 클래스 다이어그램으로 정의하고 추출부분에서 추출된 엘리먼트들의 구조를 변경한다.

추출-절단 질의와 관련된 XML-GL 요소에는 Star, ANY, Binding edge, Order가 있으며 이에 대한 사상 규칙은 다음과 같다.

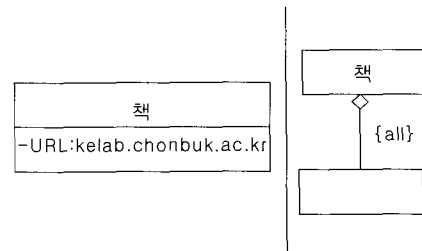
XML-GL의 Star arc는 다음 규칙 1과 같이 UML로 사상된다.

**[규칙 1]** XML-GL의 Star arc는 UML에서 {all} 제약조건을 갖는다.

예 1) 책 엘리먼트의 하위 엘리먼트 중 PCDATA를 갖고 있는 엘리먼트를 찾아라(RHS 부분).



(그림 4)는 “kelab.chonbuk.ac.kr”에 있는 문서 집합으로부터 “책” 엘리먼트를 상위 엘리먼트로 갖는 모든 문서를 찾아라.”라는 질의를 표현한 것이다.

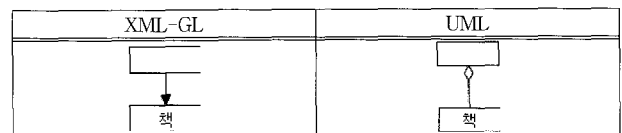


(그림 4) 추출-절단 질의의 예

XML-GL의 ANY Label은 다음 규칙 2와 같이 UML로 사상된다.

**[규칙 2]** XML-GL의 ANY Label은 UML에서 빈 클래스를 생성하며 집단화 관계를 갖는다.

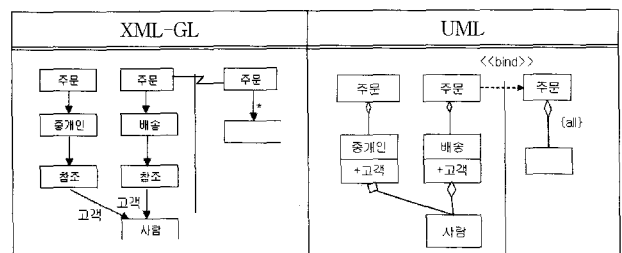
예 2) 책이라는 하위 엘리먼트를 갖는 상위 엘리먼트를 찾아라(RHS 부분).



LHS 부분에 두 개 이상의 같은 엘리먼트가 나타나고 그 중 RHS 부분의 엘리먼트와 관련 있음을 나타낼 때 사용하는 Binding Edge를 나타내는 사상시킨 규칙은 다음과 같다.

**[규칙 3]** XML-GL의 Binding Edge는 UML에서 <<bind>> stereotype이 되며 의존관계를 갖는다.

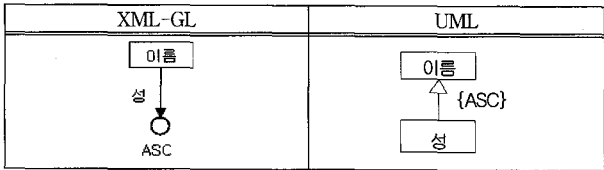
예 3) 기존에 주문한 중개인과 같은 사람에게 배송된 주문을 찾고 모든 주문 내역을 출력하라.



XML-GL의 Order는 ASC Order와 DESC Order로 구분되면 다음 규칙 4와 같이 UML로 사상된다.

**[규칙 4]** XML-GL의 ASC Order와 DESC Order는 UML에서 {ASC} 제약조건과 {DESC} 제약조건이 되며 순서의 기준이 되는 값을 상속받는다.

예 4) 이름 엘리먼트를 성에 따라 오름차순으로 정렬하라 (RHS 부분).



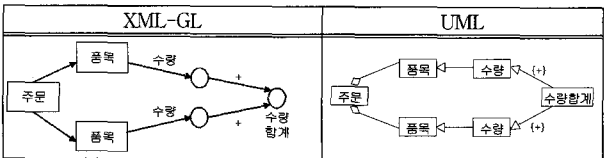
(2) 추출-일치-절단 질의

추출-일치-절단 질의와 관련된 XML-GL 요소에는 Arithmetic Arc와 Predicate Label, 그리고 Negation이 있으며 이에 대한 사상규칙은 다음과 같다.

Arithmetic Arc는 속성 값에 따른 계산과 값에 의한 계산으로 나누며 이에 따른 규칙은 규칙 5, 규칙 6과 같다.

**[규칙 5]** XML-GL의 속성값에 따라 계산되는 Arithmetic arc는 UML에서 클래스의 Properties로부터 상속받으며 속성 값에 해당하는 제약조건을 갖는다.

예 5) 주문된 품목들의 수량을 합하는 수량 합계(LHS 부분)



**[규칙 6]** XML-GL의 값에 따라 계산되는 Arithmetic arc는 UML에서 클래스의 값으로부터 상속받으며 값에 해당하는 제약조건을 갖는다.

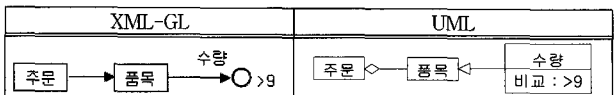
예 6) 주문한 품목의 할인율을 0.2배하여 변경(LHS 부분)



XML-GL의 Predicate는 unary와 binary로 구별한다.

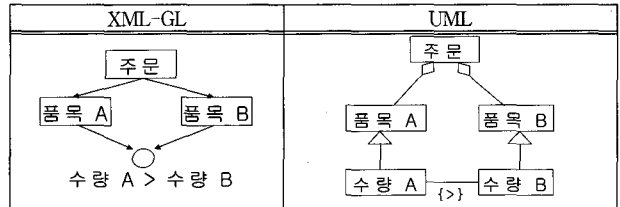
**[규칙 7]** XML-GL의 unary Predicate는 UML에서 private attribute가 된다.

예 7) 주문한 품목의 수량이 9를 초과한 경우(LHS 부분)



**[규칙 8]** XML-GL의 binary Predicate는 UML에서 private attribute가 되고 {속성 값}의 제약조건을 갖는다.

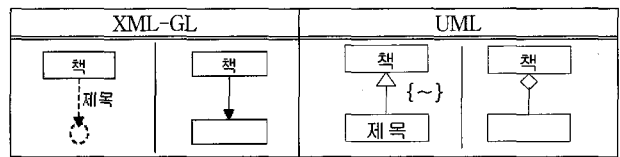
예 8) 품목 A의 수량이 품목 B의 수량보다 큰 주문(LHS 부분)



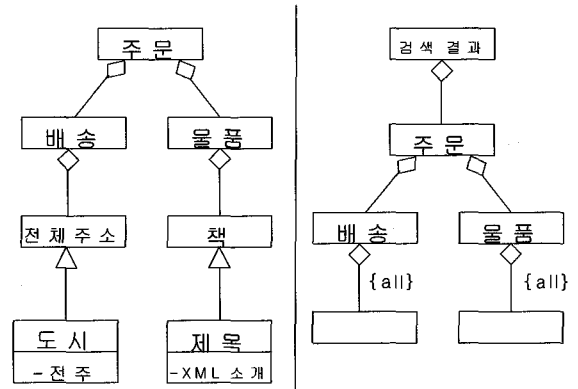
XML-GL에서 부정을 나타내는 Negation을 UML에 사상시키는 규칙은 다음 규칙 9와 같다.

**[규칙 9]** XML-GL의 Negation은 UML에서 {~} 제약조건이 된다.

예 9) 제목이 없는 책을 찾아 책 엘리먼트의 하위 엘리먼트를 출력하라.



(그림 5) “전주로 배송되었고 책 제목이 XML 소개인 주문의 배송 및 품목 내역 모두를 출력하라”는 질의를 표현한 것이다.



(그림 5) 추출-일치-절단 질의의 예

일치부분에서는 비교연산자(>, <, >=, <=, =, <>)와 문자연산자(%, %)를 사용하여 조건을 설정할 수 있다.

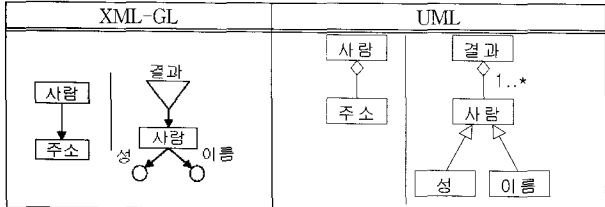
(3) 추출-일치-구성-절단 질의

추출-일치-구성-절단 질의와 관련된 XML-GL 요소에는 List와 Index가 있으며 Index는 Group\_by List 와 Aggregation

List로 구성되어 있다.

**[규칙 10]** XML-GL의 List는 UML에서 '1..\*' Multiplicity를 갖는 집단체 관계이다.

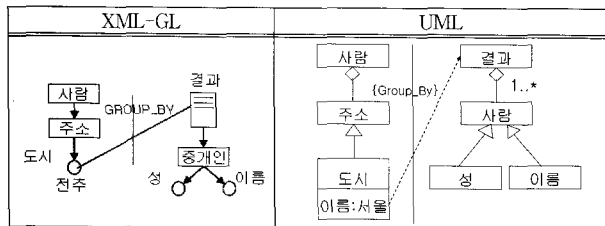
예 10) 사람의 주소에서 성과 이름을 출력하라.



XML-GL의 Index는 Group\_By와 Aggregation으로 구분하며 사상 규칙은 다음과 같다.

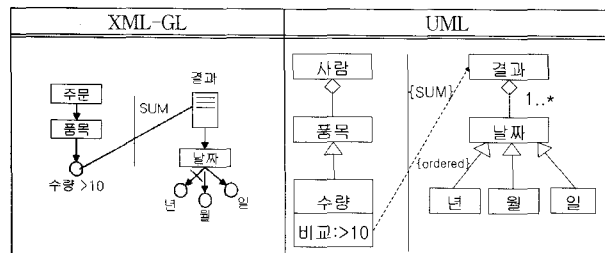
**[규칙 11]** XML-GL의 Group\_By Index는 UML에서 '1..\*' Multiplicity를 갖는 의존 관계이고 {Group\_By} 제약조건을 갖는다.

예 11) 서울이 주소인 사람을 그룹핑하여 성과 이름을 출력하라.

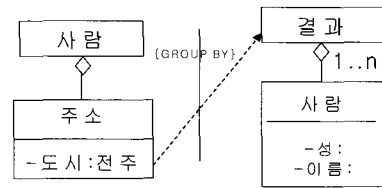


**[규칙 12]** XML-GL의 Aggregation Index는 UML에서 '1..\*' Multiplicity를 갖는 의존 관계이고, 관련 Aggregation의 종류에 따라 {SUM}, {MIN}, {MAX}, {COUNT} 중 하나의 제약조건을 갖는다.

예 12) 품목의 수량이 10 초과인 주문의 합계를 구하고 날짜를 출력하라.



(그림 6)은 “같은 도시의 사람끼리 그룹핑하여 중개인의 성과 이름을 출력하는 결과 리스트를 만들라”라는 질의를 표현한 것이다.



(그림 6) 추출-일치-구성-절단 질의의 예

#### 4.2.2 UML 클래스 다이어그램을 이용한 XML-GL 복합 질의 사상

<표 5> XML-GL 연산과 UML 사상

XML-GL 연산	UML 사상
Union	《Union》 스테레오 타입, OCL의 Union 연산
Difference	《Difference》 스테레오 타입, OCL의 excluding 연산
Cartesian Product	《Cartesian》 스테레오 타입, OCL의 select 연산
Heterogeneous Union	《Heterogeneous Union》 스테레오 타입, OCL의 exists 연산

XML-GL의 복합질의 유형에 따라, Union, Difference, Cartesian Product, Heterogeneous Union 연산으로 구분하고 복합질의 표현하기 위한 XML-GL 연산에 대한 UML에 사상시킨 규칙은 다음 <표 5>와 같다.

UML의 스테레오 타입과 OCL 제약사항을 이용하여 XML-GL에서 표현한 복합질의 UML-GL에서 지원하기 위하여 사상시킨 규칙은 다음과 같다.

##### (1) Union 연산

XML-GL의 Union 연산은 다음 규칙 13과 같이 UML에 사상된다.

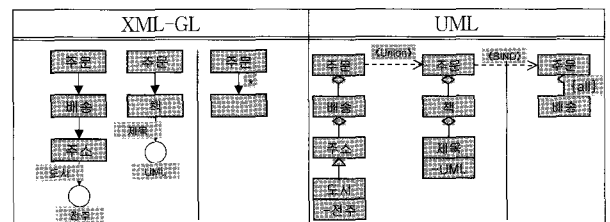
**[규칙 13]** 유니온(Union) 연산은 ‘《Union》’ 스테레오 타입으로 표현하고, <<Union>> 스테레오 타입의 제약사항은 Union 연산으로 표현한다.

UML 클래스에 대한 제약사항은 OCL로 표현한다.

##### [OCL 제약사항]

(set { A items } -> union(set { B items } = set { A items, B items})

예 13) 전주가 배송 주소인 주문과 책 제목이 UML인 주문을 찾아 하위 엘리먼트 모두를 출력하라.



OCL로 표현한 Union 연산은 다음과 같다.

set {전주가 배송주소인 세트} -> union (set {책 제목이 UML인 세트}) = set {전주가 배송주소인 세트, 책 제목이 UML인 세트}

(2) Difference 연산

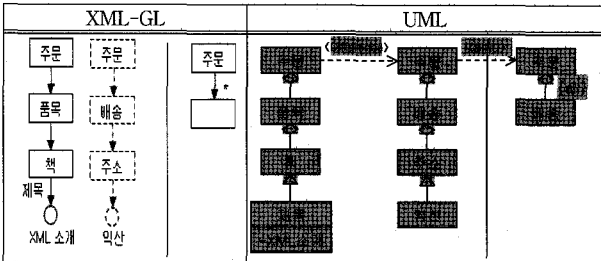
XML-GL의 Difference 연산은 다음 규칙 14와 같이 UML에 사상된다.

[규칙 14] Difference 연산은 '《Difference》' 스테레오 타입으로 표현하고, '《Difference》' 스테레오 타입의 제약사항은 'excluding' 연산으로 표현한다.

UML 클래스에 대한 제약사항은 OCL로 표현한다.

[OCL 제약사항]  
set {A items} collection->excluding(B items)=set {(A items) - (B items)}

예 14) 책 제목이 XML 소개인 주문 중에서 익산이 배송 주소인 것을 제외하고 주문 엘리먼트의 하위 엘리먼트 모두를 출력하라.



OCL로 표현한 'excluding' 연산은 다음과 같다.

set {책제목이 XML소개인 주문세트} collection->excluding(익산이 배송주소인 주문세트) = set {책제목이 XML소개인 주문세트에서 익산이 배송주소인 주문세트를 제외한 주문세트}

(3) Cartesian product 연산

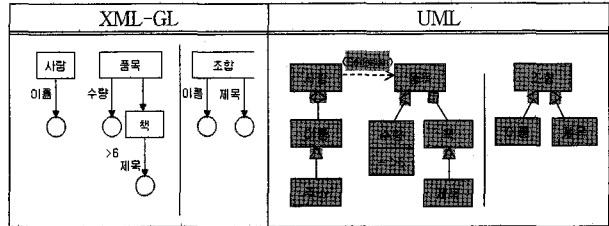
XML-GL의 Cartesian product 연산은 다음 규칙 15와 같이 UML로 사상된다.

[규칙 15] Cartesian Product 연산은 '《Cartesian》' 스테레오 타입으로 표현하고, '《Cartesian》' 스테레오 타입의 제약사항은 'select' 연산으로 표현한다.

UML 클래스에 대한 제약사항은 OCL로 표현한다.

[OCL 제약사항]  
set {A items} -> select(x | x>6) = set {6보다 큰 A items}

예 15) 이름 엘리먼트를 갖는 사람 엘리먼트와 수량이 6보다 큰 제목이 있는 책 품목 중에서 사람의 이름과 책 제목을 갖는 조합을 만들어라.



OCL로 표현한 'select' 연산은 다음과 같다.

[OCL 제약사항]  
set {제목이 있는 책 품목세트} -> select(x | x>6)=set {수량이 6보다 큰 제목이 있는 책 품목세트}

(4) Heterogeneous Union 연산

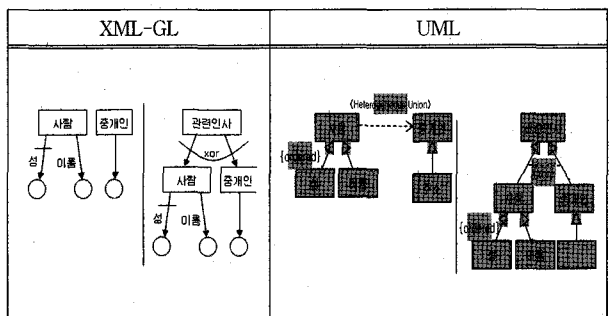
XML-GL의 Heterogeneous Union 연산은 다음 규칙 16과 같이 UML로 사상된다.

[규칙 16] Heterogeneous Union 연산은 '《Heterogeneous Union》' 스테레오 타입으로 표현하고, '《Heterogeneous Union》' 스테레오 타입의 제약사항은 'exists' 연산으로 표현한다.

UML 클래스에 대한 제약사항은 OCL로 표현한다.

[OCL 제약사항]  
set {A items, B items} -> exists {A items}=true

예 16) 성과 이름을 갖는 사람 엘리먼트와 중개인 들을 추출하고 관련인사라는 새로운 엘리먼트를 만들고 성과 이름을 갖는 사람이나 중개인 둘 중에 하나를 출력하라.





OCL로 표현한 'exists' 연산은 다음과 같다.

```
set {'성과이름을 갖는 사람세트', '중개인 세트'} -> exists {'성과이름을 갖는 사람세트'}=true
```

4.3. 사상 알고리즘

XML-GL 기반의 XML DTD와 문서 인스턴스, 질의를 입력 받아 UML 클래스 다이어그램을 생성하는 알고리즘은 [알고리즘 1]과 같다.

```
[알고리즘 1] XML-GL을 UML 다이어그램으로 사상하는 알고리즘

입력 : XML DTD, XML 문서, XML 질의
출력 : UML 클래스 다이어그램
begin { //XML Data하에서 그래픽 질의 처리
  if (Q : Query)
  then { //단순 질의 처리, DG : DocGraph
    QRT = ComputeSchema(Q.rhs); // RHS 부분 추출
    found = ExtractMatch(In, Q.lhs, bindingset, QRT)//Extract_match 처리
    if(!found)
      return emptydoc;
    Clip(DG, Q.rhs, QRT); // Clip part 처리
    Construct(Q.rhs, QRT); // Construct part 처리
    return OutputDoc(DG, QRT, Q);
  }
  else { // 복합질의 처리
    ComponentSet = DeCompose(Q); // Component Set 분리하고 재구성
    QRTset = ComputeSchema(ComponentSet) // 질의 결과 테이블 QRT 생성
    // (LHS, RHS, Binding Edge)
    Processing_Extract_Matching (QRT, Component, I)
    // 각 컴포넌트와 질의 결과 테이블과 대응시켜 간단한 질의처리
    Cartesian_Product_Computation();
    // 두 컴포넌트와 같은 RHS 갖고 있어 카디시안 연산을 수행
    for (each(QRT, Component) in QRTset)
      Clip(DG, Component.RHS, QRT);
    // 질의 결과 테이블을 이용하여 Clip part 처리
    QRTset = ComputeFinalSchema(QRTset);
    // 컴포넌트들을 머지하여 Extended Union 처리
    Extended_Union_Process(QRT1, QRT2, C1, C2);
    for(each(QRT, Component) in QRTset)
      Construct(Component.rhs, QRT);
    // 최종 질의 결과테이블 각각에 대해서 Construction 처리.
    return OutputDoc(DG, QRTset, Component);
    // 모든 질의 결과테이블에 적용하여 Output 처리
  }
}
end
```

4.4. 적용사례

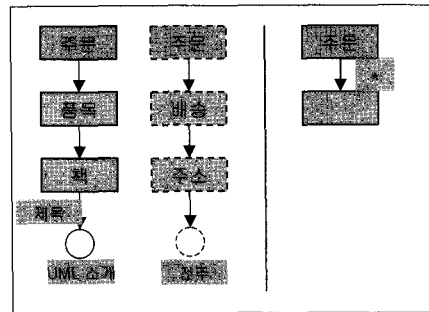
4.3절의 규칙들을 종합한 예제를 통하여 제안된 모델의 타당성을 검증한다. 또한 이를 뒷받침 할 수 있는 구현 과정은 추후 연구과정에서 제시하도록 한다. (그림 7)은 도서관 관련 XML 문서의 예를 나타낸 것이다.

예 17) 책 제목이 UML 소개인 주문 중에서 전주가 배송 주소인 것을 제외하고 주문 엘리먼트의 하위 엘리먼트 모두를 출력하라.

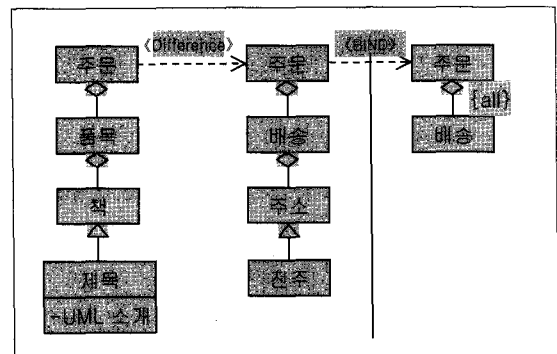
(그림 9)은 위 질의를 UML로 표현한 것이다.

<pre>&lt;?xml version="1.0" standalone="no" encoding="UTF-8"?&gt; &lt;DOCTYPE 주문 SYSTEM "주문.dtd"&gt; &lt;주문 번호=1&gt;   &lt;배송&gt;     &lt;참조 고객="C00001"&gt;&lt;/참조&gt;   &lt;/배송&gt;   &lt;중개인&gt;홍길동&lt;/중개인&gt;   &lt;품목&gt;     &lt;책&gt;       &lt;ISBN&gt;15536455&lt;/ISBN&gt;       &lt;제목&gt;XML 소개&lt;/제목&gt;       &lt;가격&gt;25000&lt;/가격&gt;       &lt;저자&gt;&lt;성&gt;김&lt;/성&gt;&lt;이름&gt;철수&lt;/이름&gt;&lt;/저자&gt;     &lt;/책&gt;     &lt;책&gt;       &lt;isbn&gt;15532155&lt;/isbn&gt;       &lt;제목&gt;Internet 소개&lt;/제목&gt;       &lt;가격&gt;22000&lt;/가격&gt;       &lt;저자&gt;&lt;성&gt;이&lt;/성&gt;&lt;이름&gt;수정&lt;/이름&gt;&lt;/저자&gt;     &lt;/책&gt;     &lt;수량&gt;10&lt;/수량&gt;     &lt;할인&gt;.30&lt;/할인&gt;   &lt;/품목&gt;   &lt;날짜&gt;&lt;년&gt;2004&lt;/년&gt;&lt;월&gt;2월&lt;/월&gt;&lt;일&gt;10&lt;/일&gt;&lt;/날짜&gt;   &lt;주문 번호=2&gt;     &lt;배송&gt;       &lt;주소&gt;&lt;회사&gt;정일서점&lt;/주소&gt;       &lt;도시&gt;전주&lt;/도시&gt;       &lt;상세주소&gt;서신동 840-2&lt;/상세주소&gt;     &lt;/배송&gt;     &lt;중개인&gt;       &lt;참조 고객="C00002"&gt;&lt;/참조&gt;     &lt;/중개인&gt;</pre>	<pre>&lt;품목&gt;   &lt;책&gt;     &lt;ISBN&gt;15532155&lt;/ISBN&gt;     &lt;제목&gt;Internet 소개&lt;/제목&gt;     &lt;가격&gt;22000&lt;/가격&gt;     &lt;저자&gt;&lt;성&gt;정&lt;/성&gt;&lt;이름&gt;최고&lt;/이름&gt;&lt;/저자&gt;   &lt;/책&gt;   &lt;수량&gt;10&lt;/수량&gt;   &lt;할인&gt;.30&lt;/할인&gt;   &lt;날짜&gt;&lt;년&gt;2004&lt;/년&gt;&lt;월&gt;2월&lt;/월&gt;&lt;일&gt;10&lt;/일&gt;&lt;/날짜&gt;   &lt;사람 사람번호="C00001"&gt;     &lt;성&gt;박&lt;/성&gt;     &lt;이름&gt;종수&lt;/이름&gt;     &lt;주소&gt;&lt;회사&gt;광진서점&lt;/회사&gt;     &lt;도시&gt;전주&lt;/도시&gt;     &lt;상세주소&gt;덕진동 1318&lt;/상세주소&gt;   &lt;/주소&gt;   &lt;사람&gt;     &lt;사람 사람번호="C00002"&gt;       &lt;성&gt;정&lt;/성&gt;       &lt;이름&gt;최고&lt;/이름&gt;       &lt;주소&gt;&lt;회사&gt;정일서점&lt;/회사&gt;       &lt;도시&gt;전주&lt;/도시&gt;       &lt;상세주소&gt;서신동 840-2&lt;/상세주소&gt;     &lt;/주소&gt;   &lt;/사람&gt;   &lt;사람&gt;     &lt;사람 사람번호="C00003"&gt;       &lt;성&gt;이&lt;/성&gt;       &lt;이름&gt;수정&lt;/이름&gt;       &lt;주소&gt;         &lt;도시&gt;익산&lt;/도시&gt;         &lt;상세주소&gt;모현동 56&lt;/상세주소&gt;       &lt;/주소&gt;     &lt;/사람&gt;</pre>
---	---

(그림 7) 도서관 관련 XML 문서의 예



(그림 8) XML-GL 질의의 예



(그림 9) UML 표현의 예

OCL로 표현한 'excluding' 연산은 다음과 같다.

```
set {책제목이 UML 소개인 주문세트} collection->excluding(전주가 배송주소인 주문세트)
=set {책제목이 UML 소개인 주문세트에서 전주가 배송주소인 주문세트를 제외한 주문세트}
```

<pre>&lt;주문&gt;   &lt;배송&gt;     &lt;주소&gt;       &lt;도시&gt;익산&lt;/도시&gt;       &lt;이름&gt;수정&lt;/이름&gt;       &lt;상세주소&gt;인화동 127&lt;/상세주소&gt;     &lt;/주소&gt;   &lt;/배송&gt; &lt;/주문&gt;</pre>	<pre>&lt;주문&gt;   &lt;배송&gt;     &lt;주소&gt;       &lt;도시&gt;익산&lt;/도시&gt;       &lt;이름&gt;수정&lt;/이름&gt;       &lt;상세주소&gt;동산동 212&lt;/상세주소&gt;     &lt;/주소&gt;   &lt;/배송&gt; &lt;/주문&gt;</pre>
--	--

(그림 10) 질의 결과의 XML 문서 표현

(그림 10)은 위의 질의에 대한 결과를 XML 문서로 표현한 것이다.

### 5. 결론 및 향후 연구 과제

본 논문에서는 WWW의 활성화와 유연하고 개방적인 속성으로 인하여 그 응용 범위가 광범위해지고 있는 XML 문서를 검색하기 위한 UML 클래스 다이어그램을 이용한 XML-GL 질의 모델링 기법을 제안하였다. 이를 위해 XML 문서와 DTD에 대한 모델링 기능과 이에 대한 시각적 질의 기능을 가지고 있는 XML-GL을 UML 다이어그램으로 사상하는 모델링 규칙과 알고리즘을 제안하였다. 또한, 모델 구성요소에 대한 제약사항을 정확하게 묘사하기 위하여 객체 제약언어인 OCL(Object Constraint Language)로 정의하였다. 이를 통해 XML 문서 질의시 객체지향 속성을 적용할 수 있었으며, 결과적으로 XML 문서에 대한 검색시 UML 클래스 다이어그램을 이용한 질의 기능을 제공하기 위한 기반을 구축할 수 있었다.

본 논문에서 제안한 UML 클래스 다이어그램을 이용한 XML-GL 질의 모델링 기법을 적용하여 XML 문서의 구조적 특징을 충분히 활용하지 못하고 있는 기존의 데이터베이스 저장 기법이나 질의 기법의 한계를 벗어나 XML 문서의 내용기반 데이터 특성과 객체지향 속성을 활용함으로써 객체지향 데이터베이스에 XML 문서를 효율적으로 저장, 관리, 검색할 수 있는 기반이 될 것이다.

향후 본 논문에서 제안한 사상 규칙과 사상 알고리즘을 객체지향 데이터베이스에 보다 체계적으로 적용할 수 있도록 하기 위하여 사상 규칙을 형식언어로 표현하기 위한 연구와 XML 문서 데이터베이스에 대한 질의를 직관적으로 시각화하기 위한 연구를 수행할 것이다. 또한 제안된 객체 모델링 기법과 질의 모델링 기법을 적용한 시각적 XML 데이터베이스 시스템을 개발하기 위한 연구도 수행되어야 한다.

### 참고 문헌

[1] Tim Bray, et al., "XML 1.0(Third Edition)," W3C Recommendation, <http://www.w3.org/TR/2004/REC-xml-20040204>, Feb. 2004.

[2] Stefano Ceri, et al., "XML-GL: a graphical language for querying and restructuring XML Documents," Computer Networks, Vol. 31, pp. 1171-1187, 1999.

[3] Stefano Ceri, et al., "Complex queries in XML-GL," Proc. of 2000 ACM symposium on Applied Computing(SAC2000), pp. 888-893, Como, Italy, Mar. 2000.

[4] Sara Comai, Ernesto Damiani, and Piero Fraternali, "Computing Graphical Queries over XML Data," ACM Transaction on Information System(TOIS), Vol. 19, No. 4, pp 371-430, 2001.

[5] Alin Deutsch, et al., "XML-QL: A Query Language for XML," <http://www.w3.org/TR/NOTE-xml-ql>, 1998.

[6] Scott Boag, et al., "XQuery 1.0: An XML Query Language," <http://www.w3.org/TR/xquery>, 2004.

[7] Sharon Adler, et al., "Extensible Stylesheet Language (XSL) Version 1.0," <http://www.w3.org/TR/xsl/>, 2001.

[8] Angela Bonifati, Stefano Ceri, "Comparative Analysis of Five XML Query Languages," ACM SIGMOD Record, Vol. 29, No. 3, pp. 76-87, 2000.

[9] V. Christophides, S. Abiteboul, S. Cluet, and M. Scholl, "From Structured Documents to Novel Query Facilities," ACM SIGMOD Record, Vol. 23, No. 2, pp. 313-324, June 1994.

[10] OMG Unified Modeling Language Specification Version 1.5, <http://www.omg.org/docs/formal/03-03-01.pdf>, Mar. 2003.



#### 최 봉 진

e-mail : korsys@empal.com  
 1997년 호원대학교 컴퓨터공학과(학사)  
 2000년 전북대학교 전산통계학과(이학석사)  
 2003년 전북대학교 전산통계학과  
 박사과정(수료)  
 2002년~현재 EAI, BPM Consultant

관심분야 : XML, EAI, BPM, SOA, ESB 등



#### 유 춘 식

e-mail : csyoo@chonbuk.ac.kr  
 1991년 전북대학교 전산통계학과(이학사)  
 1994년 전북대학교 전산통계학과(이학석사)  
 2005년 전북대학교 전산통계학과(이학박사)  
 2006년 8월~현재 전북대학교 정보전산원  
 학사행정지원팀

관심분야 : XML, 스키마 통합, 유비쿼터스, 적응형 사용자 인터페이스, W3 웹 서비스 등



#### 김 용 성

e-mail : yskim@chonbuk.ac.kr  
 1978년 고려대학교 수학과(이학사)  
 1984년 광운대학교 전산학과(이학석사)  
 1992년 광운대학교 전산학과(이학박사)  
 1985년~현재 전북대학교 전자정보공학부 교수  
 2006년 12월~현재 전북대학교 정보전산원 원장

1996년~1998년 1월 한국학술진흥재단 전문위원

관심분야 : XML, 사용자 중심의 정보검색, 다중 사용자 인터페이스, W3C 웹 서비스 등