



# XML과 데이터베이스

서울대학교 박상원 · 정재목 · 정태선 · 김형주\*

## 서론

XML은 웹 상에서 데이터 교환을 위해 제안된 표준 언어이다. 웹에서의 자유로운 데이터 교환을 위해, XML은 DTD를 통하여 문서 자체에 문서의 구조를 기술하고 있다. 이와 같이 문서의 구조는 사용자가 원하는 대로 정의할 수 있으며, 이러한 구조적 유동성은 모든 형태의 데이터가 XML로 기술될 수 있도록 해준다. 이것은 웹에서 운용되는 모든 데이터가 동일한 형태로 통합, 저장, 처리될 수 있는 기반을 제공한다.

앞으로 많은 데이터가 XML로 표현될 것이고 이런 데이터를 효율적으로 관리하기 위해서는 데이터베이스의 사용이 필수적이다. XML이 등장하기 전부터 비정형(semistructured) 데이터에 대한 연구가 데이터베이스 연구자들에 의해 활발히 연구되어 왔으며, 그 구조의 유사성으로 인하여 비정형 데이터에 대한 연구 결과를 거의 XML 데이터에 적용할 수 있다.

XML 데이터를 데이터베이스에 저장하여 질의하기 위한 방법은 많이 연구되어 왔다[1~5]. 이러한 방법들에는 XML을 위한 새로운 데이터베이스 시스템을 만들거나, 기존의 데이터베이스 시스템에 저장하는 것들이다. 기존의 데이터베이스 시스템에 저장할 때 데이터 모델의 이질성으로 인하여 이를 저장하기 위한 다양한 방법이 제시되었으며, 이는 기존의 데이터베이스 시스템의 특성을 잘 이용하여 XML 질의 수행을 빠르게 하도록 고안되었다.

본 고의 구성은 다음과 같다. 2절에서는 XML의 데이터 모델에 대하여 설명하고, 3절에서는 이러한 XML 데이터를 데이터베이스에 저장하기 위한 여러 방안들에 대하여 설명한다. 4절에서는 XML 질의어 처리에 대하여 기술하고, 5절에서는 저자들이 제안한 XML을 위한 미들웨어 시스템에 대하여 간략히 설명한다.

## 2. 데이터 모델

기존의 비정형 데이터 모델로 제안된 것은 Garlic 프로젝트에서 쓰이는 GDL과, TSIMMIS[6], Lore[5]에서 사용되는 OEM (Object Exchange Model), XWEET[7]에서 사용하는 XDM(XWEET Data Model) 등이 있다. 이러한 데이터 모델은 자기 기술적(self-describing) 모델로서 응용 프로그램들 사이의 데이터 교환에 오랫동안 사용되어 왔으며, 최근 이질적인 시스템(heterogeneous system) 사이의 데이터 교환을 위하여 널리 사용하게 되었다.

OEM은 TSIMMIS 프로젝트에서 데이터 교환을 위한 목적으로 명확히 정의되었으나, 후에 비슷한 프로젝트에서 OEM의 변종들이 사용됨에 따라 비정형 데이터 모델의 준 표준적인 역할로 사용되게 되었다. OEM은 각 객체들이 OID를 가지고 있고, 그 객체를 기술하는 텍스트 라벨과, 타입, 값을 가지고 있다. OEM은 복합(complex) 또는 단일(atomic) 타입을 가질 수 있다. 단일 타입의 예로는 자연수, 실수, 스트링, 이미지, 비디오 클립, 사운드 클립, 질의어 등이 있다. 복합 타입은 OEM 객체들의 집합을 갖는다. XML에 쓰이는 데이터 모델은 OEM과 유사한 그래프 기

\* 종신회원

반의 데이터 모델이다. 각 객체들 사이에 참조 관계가 존재하고, 그 참조 관계는 사이클이 존재할 수도 있다. XML은 OEM 모델에는 없는 하위 객체들간의 순서 개념이 있다.

XML 데이터를 접근하기 위하여 응용 프로그램은 XML 프로세서와 협력을 해야한다. XML 프로세서가 특정한 정보를 접근하기도 하고, 엘리먼트(element)의 순서를 바꾸는 작업을 수행하기도 한다. 이러한 XML 프로세서와의 통신은 API를 이용하여 이루어진다. W3C에서는 웹 브라우저와 임베디드(embedded) 스크립트가 XML과 HTML 문서를 접근하는 방법을 기술하기 위하여 DOM[8](Document Object Model)을 개발하였다. DOM은 C++, Java, Javascript등과 같은 객체 지향 프로그래밍 언어의 사용을 가정하고 있기 때문에 객체 모델로 디자인되었다. DOM에서 모든 문서(document)는 Node 클래스에 기반을 두고 있다. Node 클래스는 그것의 타입<sup>1)</sup> 등을 판단하고, 노드 구조체를 가지고 접근하고 편집할 수 있는 메소드를 가지고 있다. 그림 1과 같은 XML을 DOM으로 표현하면 그림 2와 같다.

```
<answer>
  <Book>
    <Author>Johns</Author>
    <Year>1980</Year>
    <Title>Database Theory</Title>
  </Book>
</answer>
```

그림 1 XML의 예

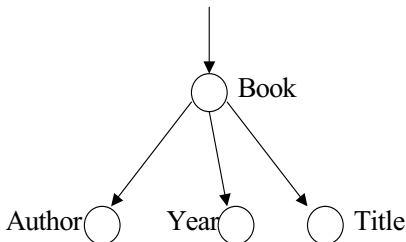


그림 2 XML을 DOM으로 표시한 예

1) Document, element, attribute, processing instruction, comment 혹은 text

### 3. XML과 데이터베이스

XML 데이터를 데이터베이스에 저장하는 것은 저장 매체, 저장 방법에 따라 여러 가지 방법으로 나눌 수 있다.

#### 3.1 데이터베이스에 따른 분류

XML을 위한 데이터베이스는 크게 두 가지 방법으로 구현할 수 있다. 첫 번째 방법으로 Lore[5]와 같이 새로운 데이터 모델을 지원하는 데이터베이스 시스템을 하부 저장 장치로부터 질의어 처리기까지 모두 만드는 것이다. XML은 기존의 관계형 데이터베이스의 릴레이션이나 객체지향형 데이터베이스의 클래스와는 다르게 그 구조가 가변적일 수 있고 질의의 형태도 정규 경로식과 같이 기존 데이터베이스 질의어와 다르기 때문에 이와 같은 데이터 모델을 잘 지원할 수 있는 것은 전용 데이터베이스 시스템을 구현하는 것이다. 즉, 이와 같은 방법의 장점은 새로운 모델을 부조화(impedance mismatch)없이 잘 표현할 수 있고 질의할 수 있다는 점이다. 그러나 [9]에서 말한 바와 같이 시스템의 안정성 및 구현 완료 가능성이 떨어진다는 것이 문제이다. 새로운 시스템을 구현하면 성능 및 안정성에 관한 확신이 있을 때까지 실제 문제에 적용하는 것이 어렵다. XML의 데이터 모델을 트리 형태로서 릴레이션을 모델로 하는 관계형 데이터베이스와는 데이터 모델이 상이하다. 그러므로 기존의 관계형 데이터베이스 시스템에서 많이 연구되어 왔던 질의어 최적화, 동시성 제어 등의 기능을 잘 활용할 수 없는 단점이 있다.

두 번째 방법은 [1~4]와 같이 기존의 데이터베이스 시스템을 이용하여 XML 데이터를 저장하고 질의를 수행하는 것이다. 이것은 XML 문서를 저장하기 위한 적절한 테이블이나 클래스를 생성한 다음 XML에 대한 질의를 데이터베이스에 대한 질의어로 변환하여 원하는 결과를 얻는 방법이다. 이렇게 함으로써 기존의 기술로 새로운 데이터 형태인 XML을 지원할 수 있게 되어 새로운 시스템을 구현하지 않고 래퍼(wrapper)를 이용하여 원하는 결과를 생성할 수 있는 장점이 있다. 기존의 데이터베이스로는 관계형 데이

타베이스와 객체지향형 데이터베이스 및 객체관계형 데이터베이스가 있다. 기존의 데이터베이스에 저장하기 위하여 스키마를 생성하는 것이 가장 중요한 점이다. 스키마를 생성하기 위한 방법은 여러 가지가 제안되었다[2, 3]. 스키마는 엘리먼트의 의미를 살려서 저장하는 방법[3]과 트리 형태의 XML 문서를 구조적으로 저장하는 방법[2]이 있다. 이것에 대하여는 다음 절에서 설명한다.

### 3.2 저장 방법에 따른 분류

저장 방법으로는 크게 두 가지 방법이 있는데 첫째로는 XML 데이터를 하나의 큰 객체로 저장하는 방법이고, 두 번째는 XML 데이터를 적절한 크기의 객체로 나누어 저장하는 방법이다. 전자는 파일이나 관계형 데이터베이스의 BLOB 혹은 객체지향형 데이터베이스의 LargeObject에 저장하는 방법이다. 이와 같은 방법은 구현이 간단하다는 장점이 있으나 XML 문서에서 원하는 정보를 추출하기 위해서는 XML 문서 전체를 메모리에 적재한 후 파싱해야 한다는 단점이 있다. 또한 이것은 갱신 연산을 수행하기가 어렵고 많은 동시 사용자가 접근하도록 하는 것에도 문제가 있다.

후자는 적절한 크기의 객체로 나누어 저장하는 방법으로서 이와 같은 방법에도 두 가지 방법이 있다. 즉, XML 문서의 의미를 보존하여 저장하는 방법[3, 4]과 구조적인 정보를 이용하여 저장하는 두 가지 방법[2]이 있다. 예를 들어 의미적인 방법은 그림 1의 XML 문서에서 Book이라는 태그를 이용하여 Book 테이블이나 클래스에 Book 엘리먼트를 저장하는 방법이다. 이것은 엘리먼트의 의미적인 정보를 유지할 수 있으므로 데이터베이스에 저장된 XML 문서에 의미있는 질의를 쉽게 작성할 수 있는 장점이 있다. 그러나 XML 문서의 DTD를 이용하여 스키마를 생성하므로 구조가 가변적이거나 확정되지 않으면 사용하기가 어렵거나 성능이 떨어진다. 여기에는 [1, 3]과 같이 적절한 테이블을 주어진 규칙에 의거하여 찾아내는 방법이 있으며, 이와는 다르게 [4]의 PathString과 같이 정규 경로식이 있는 질의를 빠르게 수행하기 위하여 모든 Path를 문자열로 저장하는 방법이 있다. 구조적인 방법은

DOM 객체의 형태를 저장하는 방법으로 엘리먼트를 저장하는 테이블이나 클래스에 Book 태그를 저장하는 방법이다. 이것은 XML 문서의 구조적인 정보를 유지하기 때문에 XML 응용 프로그램에서 적용할 때 편리한 장점이 있다. 즉 트리 형태의 XML 데이터를 테이블에 매핑하여 저장하는 것이다.

## 4. XML 질의 언어와 질의 처리 기법

이 절에서는 현재까지 제안된 XML 질의 언어와 질의 처리 기법을 다룬다.

### 4.1 질의 언어

XML 데이터는 그래프 기반의 비정형 데이터 모델에 매핑되어 질의가 수행될 수 있기 때문에 비정형 질의 언어를 XML 질의에 이용할 수 있다. 이러한 질의 언어로는 UnQL[10], Lorel[11] 등이 있다. 이 밖에도 XML에 특화된 질의 언어로는 XML-QL[12]과 XQL이 제안되었다.

먼저 XML 문서를 그래프로 표현하는 방법은 다음과 같다. O를 무한한 객체 식별자(object identity)의 집합, C를 O와 교집합을 가지지 않는 무한한 상수의 집합이라 가정하면 데이터 그래프 DB를  $DB=(V,E)$ 로 정의할 수 있다. 여기서  $V \subseteq O$ 는 노드의 집합으로 자손 에지를 포함하는 객체로서 객체 식별자로 표시된다.  $E \subseteq V \times C \times V$ 는 방향성 있는 에지(directed edge)의 집합이다.

XML 데이터는 XML의 요소(element)를 노드 V에, 엘리먼트-애틀리뷰트 관계와 엘리먼트-부엘리먼트(subelement) 관계, 엘리먼트간 참조 관계를 에지 E에 매핑하고 XML 문서의 값을 그래프의 단말 노드에 매핑하면 비정형 데이터 모델에 매핑될 수 있다. 이때, 그래프 모델은 엘리먼트-부엘리먼트 관계와 엘리먼트 간의 참조 관계(IDREF)를 구분하지 않고, 애틀리뷰트와 부엘리먼트 관계를 구분하지 않으므로 XML 문서의 특정 정보를 잃어버릴 수 있지만 간단한 데이터구조를 첨가함으로써 XML 데이터를 그래프 모델로 매핑할 수 있다. 질의는 데이터 그래프의 경로에 대한 정규 경로식(regular path expression)을 기반으로 한다. 정규 경로식은 다음과 같이 정의된다.

정의 1 정규 경로식은  $R = \epsilon | a | (R1.R2) | (R1 | R2) | R^*$ 으로 표현된다. 여기서 R, R1, R2는 정규 경로식,  $a \in C$ 는 특정 상수,  $\epsilon$ 은 빈 문자열(empty string)을 나타낸다.

이렇게 XML 질의는 정규 경로식을 바탕으로 하므로 기존 데이터베이스의 질의 언어인 SQL이나 OQL에 비하여 데이터베이스의 구조를 모르는 사용자에게 다양한 형태로 질의를 수행할 수 있도록 한다. 여기서는 실제 XML-QL의 예를 통하여 XML 질의의 사용 예를 보인다. XML-QL은 XML을 위한 질의 언어로서 XML 데이터에 대하여 질의를 수행하고, 새로운 XML 형태의 문서를 만들 수 있도록 고안된 언어이다. XML-QL은 패턴 매칭에 기반한 질의 언어라는 특징을 가진다. 다음은 bib.xml 문서에서 brand가 Ford인 논문을 찾아 그 저자를 출력하는 질의이다.

```
where <$*> <name> $r </name>
      <SE> Ford </> in "www.a.b.c/bib.xml",
      SE in {brand}
```

```
construct <result> $r </result>
```

여기서 \$는 어떠한 예제에도 매핑 가능함을 나타내고, \*는 예제가 하나도 나타나지 않을 수도 있고 하나 이상 나타날 수도 있음을 나타내는 정규 경로식으로 \$\*는 임의의 경로가 올 수 있음을 나타낸다.

## 4.2 질의 처리

XML 데이터에 대한 질의가 주어졌을 때 질의를 처리하는 방법은 크게 기존 데이터베이스 시스템을 이용하는 방법과, XML 데이터를 비정형 데이터의 한 인스턴스로 보고 비정형 데이터를 위해 고안된 질의 엔진을 이용하는 두 가지 방법으로 나눌 수 있다. 비정형 데이터에 대한 질의 처리는 그래프 탐색이 기본으로 다음의 세 가지 방법을 생각할 수 있다[13].

1. 하향식 방법(top-down method): 주어진 경로식에 대하여 순 방향으로 객체를 변수에 바인딩 시키는 방식이다.
2. 상향식 방법(bottom-up method): 하향식 방법과는 반대 방향으로 객체를 변수에 바인딩 시키는 방식으로 특정 선택 조건(selection condition)을 만족하는 부모 경로에 대한 인덱스를

가지는 Vindex[13] 등으로 구현될 수 있다.

3. 복합 방법(hybrid method): 하향식 방법(top-down method)과 상향식 방법(bottom-up method)을 조합한 방식이다.

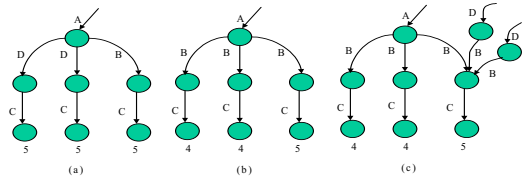


그림 3 다양한 형태의 데이터 그래프

예를 들어, 그림 3과 같은 그래프 형태에서 `select x From A.B x where exists y in x.C:y=5`의 질의를 처리한다고 하자.

그림 3에서 (a)의 경우에는 경로식 A.B를 만족하는 경로가 적으므로 하향식 방법이 유리하다. 만약 상향식 방법으로 처리한다고 하면 단말 노드에서 모든 노드가 `y=5`의 조건을 만족하므로 불리하다. 그림 (b)의 경우에는 단말 노드 조건을 만족하는 객체가 하나밖에 없으므로 하향식 방법이 유리하다. (c)의 경우에는 그래프의 윗 부분은 상향식 방법, 그래프의 아랫 부분은 하향식 방법을 사용하는 것이 유리하다. 이렇게 데이터베이스의 통계정보에 따라 여러 가지 질의 계획(query plan)을 생성한 후 비용 계산을 통한 최적의 계획을 찾는 기법이 [13]에서 제안되었다.

이 밖에 비정형 데이터 처리에 있어서 제안된 기법은 주로 데이터로부터 스키마를 뽑아내는 시도이다. 즉, 방대한 양의 데이터 그래프를 탐색하기 보다 데이터 그래프로부터 스키마 그래프를 추출하여 이것을 인덱스로 사용하면 그래프 탐색 범위를 많이 줄일 수 있다[14~16]. 여기서는 이들 기법 중에서 1-index[16] 기법을 소개한다.

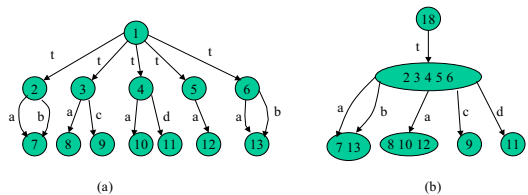


그림 4 데이터 그래프와 1-index

1-index 기법은 루트 노드에서 특정 경로식을 만족하는 객체를 효율적으로 찾는 기법이다. 그림 4는 예제 데이터 그래프(a)와 해당 1-index (b)를 나타낸다. 1-index에서 각 노드는 데이터 그래프에서 루트 노드에서 특정 노드로의 경로로 이루어지는 언어가 같은 노드의 집합을 익스텐트(extent)로 가진다. 예를 들면 데이터 그래프 (a)에서 노드 7과 노드 13은 경로식 t.a와 t.b로 도달되므로 (b)에서 같은 그룹에 속하게 된다. 이렇게 구성된 1-index는 데이터 그래프에 비하여 그 크기가 작으므로 질의 처리에 효율적이다.

### 5. 이질 환경을 위한 XML 미들웨어 시스템: XWEET

최근 들어 기존에 존재하는 많은 서로 다른 형태의 데이터를 통합, 저장, 접근하고자 하는 노력들이 급속히 늘어나고 있다. 또한 웹을 중심으로 한 많은 응용들이 등장하고 있다. 이때 웹상에서 정보 전송 및 처리에 사용되는 데이터는 XML이 될 것이다. 이에 서울대학교 컴퓨터공학부에서는 이질 환경을 위한 XML 미들웨어 시스템으로 XWEET[7]를 제안하였다. XWEET 시스템은 여러 정보 소스에 존재하는 이질적인 구조의 데이터를 단일한 구조(XML)로 표현, 관리, 접근할 수 있는 미들웨어 시스템이다.

그림 5는 전체적인 XWEET 구조를 보여주고 있다. Wrapper는 여러 외부 데이터 소스<sup>2)</sup>를 읽어 관심있는 정보만을 추출하여 XML 문서로 변

환하는 모듈이다. Data source는 XWEET 에서 사용되는 여러 XML 문서(또는 동등한 DOM 구조)를 받아 XDM(XWEET Data Model)로 변환하는 작업을 수행한다. XDM은 DOM의 간략화된 모델이다. 데이터 소스로는 여러 외부 데이터 소스로부터 XML 문서를 추출하는 래퍼, XWEET 자체적인 XML 데이터 저장소 PDM, 그리고 일반 화일 및 HTTP를 통해 얻는 XML 문서가 있다.

PDM(Persistent Data Manager)는 XWEET에서 직접 관리하는 XML 데이터들을 다루기 위한 모듈로 XML 데이터를 RDBMS에 효율적으로 저장/조작하는 기능을 제공한다. 또한 신속한 XML 데이터 검색을 위한 인덱스도 제공한다. XSI(XWEET Semantic Integrator)는 사용자로부터 받은 정보를 바탕으로, 여러 데이터 소스를 통해 받은 XML의 의미적으로 동일하지만 다른 방식으로 인코딩된 XML 문서를 동일한 뷰로 볼 수 있도록 하는 기능을 제공하는 모듈이다. 필요에 따라 입력 XML 문서의 내용을 변경하기도 한다. 중개자(mediator)는 XSI와 협력하여 서로 다른 문법과 의미(semantics)을 사용하는 XDM 모델을 질의가 가능한 동일한 형태로 변환하는 일을 한다. XQP(XWEET Query Processor)는 데이터 소스로부터 받은 여러 XML 데이터에 대한 선언적 질의를 처리하는 모듈이다. Ontology Manager는 스키마 통합이나 특정 응용 도메인에서 사용되는 용어들에 대한 통합 및 재정리를 담당한다. Application Modules는 XWEET에서 제공하는 여러 서비스를 바탕으로 응용 프로그램 고유의 작업을 처리하는 모듈들이다. WPG(Web Page Generator)는 웹 클라이언트에서 전달되는 HTTP 요청을 바탕으로 웹 페이지를 생성하는 모듈로, CGI, Servlet 등으로 구현될 수 있다. HTML/XML Templates는 WPG의 한 종류로 HTML/XML template을 통해 웹 페이지 생성을 보다 손쉽게 하는 모듈이다. XWEET Web Service Manager는 특정 응용에 속한 WPG들 사이의 데이터 및 제어 흐름을 관리하는 모듈이다.

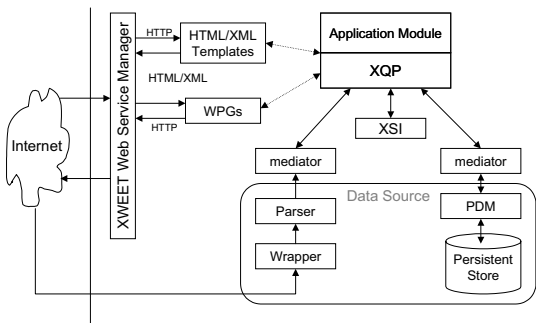


그림 5 XWEET 구조

2) Web 상의 여러 종류의 HTML 파일, e-mail, news, XML 파일 등

### 6. 맺음말

본 고에서 XML을 데이터베이스에서 지원하는

방법에 대하여 살펴보았다. XML 문서는 전용 데이터베이스 시스템을 이용하여 저장할 수도 있고, 기존의 데이터베이스 시스템을 이용하여 저장할 수도 있다. 기존의 데이터베이스 시스템을 이용하면 이미 개발된 기술을 이용할 수 있는 장점이 있지만, 데이터 모델의 부조화로 인하여 부가적인 연산이 더 필요하게 된다. XML 문서를 저장하기 위하여 적절한 스키마를 생성하여야 하며, XML 질의를 실제 데이터베이스의 질의로 변환해야 한다. XML 질의어는 기존 질의어와는 다른 점으로 정규 경로식을 사용할 수 있다는 점이다. 이러한 정규 경로식을 지원하기 위하여 여러 인덱스들이 제안되었다. XWEET는 웹 상에서 XML 응용 프로그램을 지원하기 위한 시스템으로 XML 데이터를 저장하거나 랩퍼를 통하여 XML 문서로 변환하는 작업을 수행하며, 웹 응용 프로그램을 작성할 수 있는 환경을 제공한다.

### 참고문헌

- [1] Alin Deutsch, Mary Fernandez, and Dan Suciu. Storing Semistructured Data with STORED. SIGMOD, 1999.
- [2] Daniela Florescu and Donald Kossman. Storing and Querying XML Data using an RDBMS, Data Engineering Bulletin, 22(3), September 1999.
- [3] Jayavel Shanmugasundaram, Kristin Tufte, Gang He, Chun Zhang, David DeWitt, and Jeffrey Naughton. Relational Databases for Querying XML Documents: Limitations and Opportunities. VLDB, 1999.
- [4] Takeyuki Shimura, Masatoshi Yoshikawa, and Shunsuke Uemura, Storage and Retrieval of XML Documents using Object-Relational Databases, DEXA, 1999.
- [5] Jason McHugh, Serge Abiteboul, Roy Goldman, Dallon Quass, and Jennifer Widom. Lore: A Database Management System for Semistructured Data. SIGMOD Record, 26(3), 1997.
- [6] H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and Jennifer Widom. Integrating and Accessing Heterogeneous Information Sources in TSIMMIS. In Proceedings of the AAAI Symposium on Information Gathering. pages 61-64, 3 1995.
- [7] JaeMok Jeong, Sangwon Park, Tae-Sun Chung, and Hyung-Joo Kim, XWEET: XML DBMS for Web Environment, The First SNU Workshop on Computer Science and Engineering 2000, June 16-17, 2000, Seoul, Korea.
- [8] W3C. Document Object Level(DOM) Level 1 Specification, <http://www.w3c.org/TR/>, Oct 1998.
- [9] Michael Stonebraker and Paul Brown. Object-Oriented DBMS's Tracking The Next Great Wave. Morgan Kaufmann, 2nd Ed., 1999.
- [10] Peter Buneman, Susan Davidson Gerd Hillebrand Dan Suciu, A query language and optimization techniques for unstructured data, In Proceedings of ACM-SIGMOD International Conference on Management of Data , 1996.
- [11] S. Abiteboul, Dallon Quass, Jason McHugh, Jennifer Widom, Janet Wiener, The Lorel Query Language for Semistructured Data, International Journal on Digital Libraries, 1996.
- [12] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu, Query Language for XML", In Proceedings of Eighth International World Wide Web Conference, 1999.
- [13] J. McHugh and J. Widom, Query optimization for XML, In Proceedings of the Conference on Very Large Data Bases, 1999.
- [14] Mary Fernandez, Dan Suciu, Optimizing Regular Path Expressions Using Graph Schemas, In Proceedings of the International Conference on Data Engine-

ering , 1998.

- [15] Peter Buneman, Susan Davidson, Mary Fernandez, and Dan Suciu, Adding structure to unstructured data, In Proceedings of the International Conference on Database Theory, 1997.
- [16] Tova Milo, and Dan Suciu, Index Structures for Path Expressions, In Proceedings of the International Conference on Database Theory, 1999.

**정 재 목**



1994 서울대학교 계산통계학과(학사)  
 1996 서울대학교 전산학과(석사)  
 1996~현재 서울대학교 컴퓨터공학부 박사과정  
 관심분야: 데이터베이스, XML, 트랜잭션, Web  
 E-mail: jmjeong@oopsla.snu.ac.kr

**정 태 선**



1995 KAIST 전산학과(학사)  
 1997 서울대학교 전산학과(석사)  
 1997~현재 서울대학교 컴퓨터공학부 박사과정  
 관심분야: 데이터베이스, XML, 질의 언어  
 E-mail: tschung@oopsla.snu.ac.kr

**박 상 원**



1994 서울대학교 컴퓨터공학과(학사)  
 1997 서울대학교 컴퓨터공학과(석사)  
 1997~현재 서울대학교 컴퓨터공학부 박사과정  
 관심분야: 데이터베이스, XML, Semistructured data, Web  
 E-mail: swpark@oopsla.snu.ac.kr

**김 형 주**



1982 서울대학교 전자계산학과(학사)  
 1985 Univ. of Texas at Austin (석사)  
 1988 Univ. of Texas at Austin (박사)  
 1988.5~1988.9 Univ. of Texas at Austin. Post-Doc  
 1988.9~1990.12 Georgia Institute of Technology(부교수)  
 1991.1~현재 서울대학교 컴퓨터공학부 교수  
 E-mail: hjk@oopsla.snu.ac.kr

**• 제3회 한국소프트웨어공학 학술대회 •**

- 일 자 : 2001년 2월 8 ~ 10일
- 장 소 : 강원도 피닉스파크
- 주 최 : 소프트웨어공학연구회
- 문 의 처 : 숭실대학교 컴퓨터학부 양승민 교수  
 Tel. 02-820-0912