관계형 데이터베이스 기반의 XML Schema 응용을 위한 설계 방법론

임종선, 주경수
*순천향대학교 정보기술공학부
e-mail:ronmer@chol.com, jsoojoo@sch.ac.kr

Design Methodology for XML Schema Application based on RDB.

Lim Jong-Seon, Joo Kyung-Soo
Dept of Computer Engineering, Soonchunhya-sung University

요 약

B2B 전자상거래와 같이 XML을 이용한 정보교환이 확산되고 있으며, 이에따라 상호 교환되는 정보에 대하여 체계적이고 안정적인 저장·관리가 요구되고 있다. 이를 위해 XML 응용과 데이터베이스 연계를 위하여 다양한 연구가 XML DTD와 관계형 데이터베이스 중심으로 수행되었다. 그러나, XML DTD가 다양한 데이터 타입 및 표현의 한계 때문에 관계형 데이터베이스와 원활한 연계가 불가능하게 되었다. 그러므로 다양한 데이터 타입과 데이터 표현에 강력한 XML Schema에 대한 연구가 필요하게 되었다. 또한 계층구조를 2차원 정보로 변환하는 방법에 의해서 각 구조화된 정보를 관계형 데이터베이스로 저장하기 위한 데이터 모델링과 XML schema 모델링이 요구된다.

본 논문에서는 XML schema로 정의된 XPDL 정의 스키마를 UML로 변환하는 방법론과 UML을 RDB 스키마로 변환하는 방법론을 제시함으로써, 워크플로우 정의 언어인 XPDL 정의 스키마를 토대로 관계형 데이터베이스의 스키마를 확정하는 방법론을 제안하였다.

1. 서론

XML은 구조화된 정보를 포함하고 있는 문서들을 위한 마크업 언어이다. 구조화된 정보는 구체적인 내용과 그 내용이 수행해야할 역할을 포함하고 있다. XML schema에 대한 연구는 W3C Recommendation으로 채택된 XML W3C XML Schema가 최종 검토 단계로 진행되고 있어 표준이 될 가능성이 가장 높다. W3C XML Schema는 XML DTD보다 다양한 데이터 타입을 정의할 수 있고, 강력한 표현력을 이용하여 다양한 어플리케이션으로 사용하기에 편리한 장점을 가지고 있다.

한편 XML 어플리케이션과 데이터베이스 시스템 사이의 원활한 연계를 위해서, 그동안 XML DTD를 관계형 데이터베이스 스키마로 변환하는 방법에 대해서는 많은 연구가 진행되었다. 그러나 UML 클래스를 이용하여 W3C XML Schema를 모델링하고, 그 모델링으로 교환되는 데이터를 관계형 데이터베이스로 저장하는 통합 설계 모델링에 대한 연구는 미비한 실정이다. XML이 컨텐츠에서 데이터베이스로까지 그 적용분야가 확장되면서 XML로 표현된 정보들을 어떻게 효율적으로 저장하고 관리할 기술도 큰 이슈중의 하나이다[3][4].

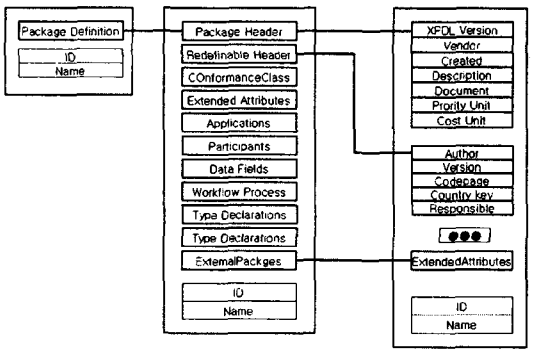
본 논문에서는 XML 모델링을 관계형 데이터 모델링으로 변환하는 데 있어서, 기존의 Ronald Bourrent의 방법을 적용하여 XML 스키마를 UML 객체 모델로 변환한 후, UML 객체 모델을 관계형 데이터베이스데이터 모델링으로 변환하는 과정을 보여준다. 본 논문에서 사용한 XML 스키마는 워크플로우 정의 언어인 XPDL(XML Process Definition Language) 정의 스키마를 토대로 하였으며, 이는

차후 XPDL 문서를 저장하기 위함이다. 본 논문에서는 XPDL 문서 저장의 선행연구로서 XML 스키마를 통한 관계형 데이터베이스 스키마 설계 방법을 다루었다. 본 논문의 2장에서는 관련연구 및 기술을, 3장에서는 XPDL 정의 스키마의 변환방법을 다루며, 마지막으로 4장에서는 결론을 기술한다.

2. 관련연구

2.1 XPDL(XML Process Definition Language)

XPDL은 XML 기반인 데이터 전송방식이며, XPDL을 통한 데이터 전송의 주요 요소들 중의하나는 다양한 툴들에 의해 사용된 정보들을 처리할 수 있는 확장성이다. 워크플로우 프로세스 정의를 표시한 제한된 속성들에 기반을 둠으로써 XPDL은 서로 다른 접근을 지원한다. 또 다른 XPDL의 중요한 요소는 공통 표현을 위해 사용되는 기업 특정 속성들을 지원하는 일반적인 구조로써 메타-모델을 묘사하고, 프로세서 정의 내에서 포함된 객체들과 속성들을 정의함으로써 XPDL 문법은 직접적으로 이런 객체들과 속성으로 관련된다[1][2].



<그림 1> XPDL의 구조

XPDL의 중요한 이슈는 작업의 적절한 분산에 있다. Activity 어트리뷰트는 동작을 실행하는데 필요한 리소스를 명세하고, 이것은 리소스의 요구를 결정하는데 필요한 경우 실행 시간을 구하는 표현이다. Activity 어트리뷰트는 애플리케이션을 실행하는데 필요한 동작을 명세한다. 이 컨셉은 리소스 개념, 애플리케이션간의 합동, 동작의 실행을 통합하여 지원한다.

XPDL에 정의된 주요 요소들을 <그림 1>과 같다. 이 주어진 요소는 XML 스키마를 이용하여 문서의 구조를 정의함으로써 XML 파서를 통하여 XML을 검증하고 처리한다. 즉 데이터 형식이나 여러 속성값, 반복 횟수 등과 같은 검정 기능에 해당하는 규칙들을 내장시킴으로써 문법적인 타당성을 검증한다.

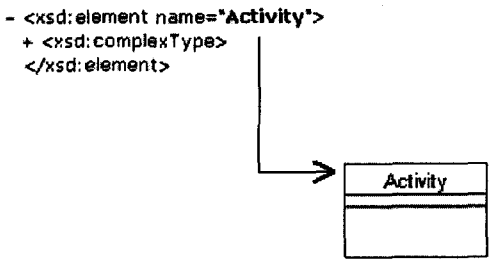
3. 변환 방법

3.1 XPDL 정의에서 UML로 변환

이 작업은 워크플로우 프로세스를 정의하는 XPDL정의를 UML 클래스로 변환하는 4가지의 방법을 제시한다. XPDL 정의(version 1.0)은 XML 스키마로 정의되어있기 때문에 본 장에서는 XML 스키마를 토대로 변환 하는 방법을 다룬다.

XML element를 UML class로 변환

UML 클래스는 객체 지향 모델에 있어서 구조적이고 행동에 관한 특징을 보여준다. 그런 특징은 attribute, association, aggregation, composition을 포함한다. 다른 측면으로, XML 엘리먼트는 어트리뷰트와 자식 엘리먼트를 포함하는 것을 제공한다. 그러므로 XML 엘리먼트와 UML 클래스는 직접적으로 매핑된다. <그림 2>에서는 XML element와 UML class 다이어그램 간의 매핑을 보여준다.

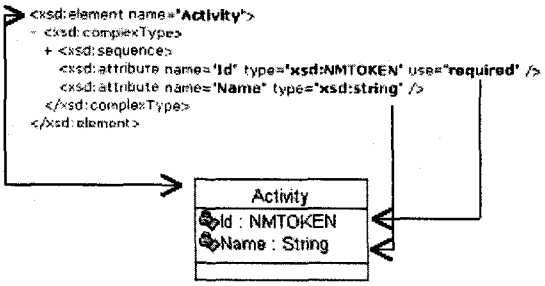


<그림 2> XML element를 UML class로의 매핑

XML element와 attribute를 UML attribute로 변환

근본적으로, 이전의 원시 데이터 타입이나 UML 어트리뷰트의 목록은 XML 어트리뷰트로 표현된다. 반면, XML 파서는 모든 추가적인 공백 문자, 탭, 라인피드 등을 제거한다. 이런 것은 주로 XML 어트리뷰트를 짧은 문자열 값의 간단한 데이터에 대한 접근을 제공한다. 다른 측면으로, 클래스에 수반되는 XML 엘리먼트의 분할된 자식 엘리먼트는 UML 클래스의 어트리뷰트로 변환된다. <그림 3>은 XML

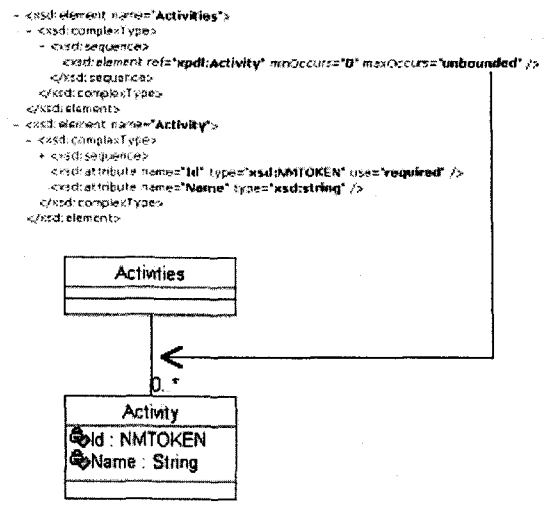
엘리먼트와 어트리뷰트를 UML 클래스와 그 클래스의 어트리뷰트로 매핑시키는 과정이다.



<그림 3> UML class와 어트리뷰트로의 매핑

XML element를 UML 객체간의 관계로 변환

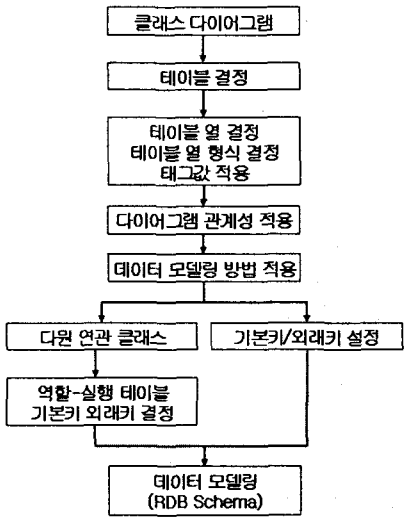
XML 스키마의 현재 버전(version 1.0)은 완벽한 객체 지향적인 모델의 표현에 대해 직접적이고 완벽한 지원을 가지지 못한다. 특히, 모델에 일반적으로 존재하는 대표(delegation)와 집합(aggregation) 사이의 구별을 지원하지 못한다. 그러므로 XML 스키마에 나타나지 않은 대표와 집합은 사용하지 않는다.



<그림 4> XML element를 UML 객체간 관계로 매핑

3.2 UML class 다이어그램에서 RDB 스키마로의 변환

UML 클래스를 이용한 데이터 모델링 과정의 다이어그램은 <그림 5>와 같다.



<그림 5> 데이터모델링의 과정

<그림 5>에서 관계형 데이터베이스 스키마 도출을 위한 데이터 모델링의 변환 방법은 다음의 방법을 사용한다[5].

① 클래스는 테이블이 됨.
② 클래스의 속성은 테이블의 열이 됨.
③ 클래스의 속성 타입은 테이블의 열 타입이 됨.
④ 속성에 {nullable} 태그가 있으면 테이블 속성에 NULL 또는 NOT NULL을 추가함.
⑤ 속성이 초기치 값을 가지면, 열에 DEFAULT 문을 추가함.
⑥ 루트나 독립적인 클래스와 같이 일반화가 없는 클래스를 위해서는 integer 기본키를 생성하고, {oid}를 위해서는 기본키 제약조건에 {oid} 태그 열을 추가함.
⑦ 자식 클래스(Subclass)들은, 각 부모 클래스의 키를 기본키와 외부키 제약조건에 추가함.
⑧ 연관 클래스들은 각 역할-실행 테이블에 대한 기본키를 기본키와 외래키 제약조건에 추가함.
⑨ 만일 {alternate oid = <n>} 태그이면, UNIQUE 제약조건에 대한 열을 추가함.
⑩ 각 명시된 제약에 대해 CHECK를 추가함.
⑪ 0..1, 1..1 규칙의 연관 관계에서 참조하는 테이블에 외부키를 생성함.
⑫ 집합 테이블(CASCADE와 같은)의 외부키를 갖는 복합 집합을 위해서 기본키를 생성한다: 기본키를 위해 추가적인 열을 추가함.

⑬ 이원 연관 클래스를 적당한 "N" 쪽 테이블로 이동함으로써 최적화 함.
⑭ 연관 클래스가 아닌 3원 연관은 N:N에 대한 테이블을 생성함.
⑮ N:N, 3원 연관에서 역할-실행 테이블의 키로부터 기본키와 외부키 제약조건을 생성함.

아래의 <그림 6>은 UML class 다이어그램을 이용하여 관계형 데이터베이스의 스키마를 도출한 결과이다. 지면 상 전체의 스키마를 보여주지 못하지만, 간단하게 요약하여 Activities UML 클래스와 Activity UML 클래스의 관계를 나타내었다.

```
Create table Activities
(
  Activities_id              int,
  Activity_Id                int,
  PRIMARY KEY                (Activities_id),
  CONSTRAINT                 ToActivity
  FOREIGN KEY (Activity_Id) REFERENCES Activity
)

Create table Activity
(
  Activity_Id                int,
  Activity_name              string,
  Description_Id             int,
  Limit_id                   int,
  Performer_id               int,
                .
                .
  ExtendedAttributes_id      int,
  PRIMARY KEY                (Activity_id),
  CONSTRAINT                 ToDescription
    FORIGEN KEY (Description_id)
    REFERENCE Description,
  CONSTRAINT                 ToLimit
    FOREIGN KEY (Limit_id) REFERENCE Limit,
                .
                .
)
```

<그림 6> UML로부터 RDB 스키마 도출

4. 결론 및 향후 연구방향

XML 어플리케이션과 데이터베이스 시스템 사이의 원활한 연계를 위해서, 그동안 XML DTD를 관계형 데이터베이스 스키마로 변환하는 방법에 대해서는 많은 연구가 진행되었다. 그러나 UML 클래스를 이용하여 W3C XML Schema를 모델링하고, 그 모델링으로 교환되는 데이터를 관계형 데이터베이스로 저장하는 통합 설계 모델링에 대한 연구는 미비한 실정이다. XML을 이용한 워크플로우 정의 언어인 XPDL은 기존에는 XML DTD를 이용하여 정의를 하였지만, XML DTD의 표현 능력이 XML 스키마보다 현저히 낮기 때문에 XML 스키마를 이용한 정의 모델이 제안되었다.

본 논문은 이러한 XML 스키마 기반의 XPDL을 UML 객체 지향 모델로의 매핑후, RDB 테이블 스키마로 변환하는 매핑 방법론을 제공하였다. 이를 이용하여 워크플로우 정의 언어인 XPDL을 이용하여 데이터베이스에 저장하는 방안을 모색할 수 있을 것이다.

향후 연구 방향으로는 본 논문의 방법론을 이용하여 워크플로우 프로세스를 위한 XML 저장소를 구현하고자 한다.

참고문헌

[1] Workflow Process Definition Language, XML Process Definition Language Document Number WFMC-TC-10525 Document Status, XPDL 1.0 beta, 2002

[2] Chengfei Liu, Hui Li, and Maria E Orlowska. Object-Oriented Design of Repository for Enterprise Workflows. CRC for Distributed System Technology and Computer Science Department, The University of Queensland, 1996. eai/concept/eai_term4.html ZDNet Korea, ZDNet Korea

[3] Minrong Song, Jhon A. Miller, " REPOX:An Repository for Workflow Designs And Specifications", August 26, 2001

[4] Workflow Management Coalition. Workflow Standards-Interoperability Wf-XML Binding. Document Number WFMC-TC-1023, pp.8-28, May 2000.

[5] 방송윤, 주경수, "데이터베이스 기반의 XML 응용을 위한, UML을 이용한 통합 설계 방법론", 정보관리학회, 19권 2호, pp 49-67, 2002

[6] 조창선, "전사적 엔터프라이즈 통합으로 진화하는 EAI", see http://www.itmast.com