

## XML 스키마로 작성된 스키마 문서의 브라우징

염지현<sup>o</sup> 김혁만

국민대학교 전산과학과 멀티미디어 데이터베이스 연구실  
{jhyum, hmkim}@kookmin.ac.kr

### Browsing schema documents written in XML Schema

Jihyun Yum<sup>o</sup> Hyeokman Kim

Multimedia Database Lab., Dept. of Computer Science, Kookmin University

#### 요 약

본 논문은 XML 스키마 문서에서 정의된 컴포넌트 요소들의 타입-엘리먼트 복합 계층 구조와 타입 계층 구조를 그래피컬하게 파악할 수 있는 XML 스키마 브라우저를 구현한 것이다. XML 스키마 브라우저의 설계 및 구현 방법을 제안하고, 그 기능을 서술한다. 구현된 XML 스키마 브라우저를 사용함으로써 복잡한 XML 스키마 문서의 구조를 한눈에 쉽고 정확하게 파악할 수 있다.

#### 1. 서 론

최근 인터넷에서의 데이터 교환 및 문서 포맷의 표준으로 XML(Extensible Markup Language)이 각광을 받고 있다[1]. 기존의 문서들은 단순한 스트링을 나열한 형태를 띠고 있는 반면, XML 형식으로 정의된 문서는 태그를 통해 의미 단위 별로 계층화 되어 있다. 이러한 의미 단위 및 계층 구조는 XML Schema 언어를 이용하여 정의된다. 스키마는 문서의 구조를 정의할 때, 복합 계층구조 (composition hierarchy) 및 타입 계층구조 (type hierarchy)와 같은 객체 지향 모델링(modeling) 기법을 사용하여 문서의 복잡한 구조도 효과적으로 정의하는 강력한 기능을 제공한다. 그러나 XML Schema에 익숙한 사용자도 XML Schema로 작성된 복잡한 스키마 문서의 전체 구조를 파악하기란 쉽지 않다. 특히 MPEG-7[2] 스키마와 같이 천여 개 이상의 타입과 엘리먼트로 구성되거나, MPEG-21과 같이 열 개가 넘는 스키마 문서들이 임포트(import) 또는 인클루드(include) 메커니즘을 사용하는 대규모 스키마의 경우, 스키마 문서의 전체 구조를 파악하기 매우 어려울 뿐만 아니라 많은 시간과 노력이 필요하다.

XML Schema로 작성된 스키마 문서의 구조를 브라우징(browsing) 할 수 있는 방법에 대한 연구가 이루어지고 있으며, 몇몇 소프트웨어도 국내 외에 출시되어 있다. 예를 들어, 스키마를 비주얼하게 표현한 SONG의 제안 방법을 살펴 보면[4], 스키마의 정의를 하향식(top-down) 구조의 다이어그램으로 표현한다. 엘리먼트를 상자로 표현하고 각 엘리먼트의 관계를 선으로 연결하여 나타낸다. 이 방법의 제한은 엘리먼트간의 복합 계층구조를

하향식 구조로 표현하여 문서 구조의 이해에 혼돈을 줄 수 있으며, 타입 계층 구조는 파악할 수 없다. 또한, Altova에서 구현한 XMLSpy[5]는 스키마를 그래피컬하게 잘 표현하고 있지만, 임의의 타입과 그 자식 엘리먼트와의 복합 계층 구조만을 파악할 수 있는 제한이 있다.

본 논문은 선행 연구의 취약점인 스키마 문서의 복합 계층구조와 타입 계층구조를 그래피컬하게 표현하고 그 구조를 브라우징 할 수 있는 XML 스키마 브라우저(이하 스키마 브라우저)를 설계 및 구현하였다.

#### 2. XML 스키마 브라우저 설계 목표

스키마 문서는 타입-엘리먼트 복합 계층구조와 타입 계층구조 형태로 이루어진다. 스키마 컴포넌트 요소 중 복합 타입(complex type)이 자식 엘리먼트(child element)들을 정의 할 경우, 수평적인 타입-엘리먼트 복합 계층 구조가 성립된다. 그리고, 복합 타입을 포함한 모든 타입은 임의의 상위 타입(supertype)의 정의를 상속 받아 새로운 애트리뷰트나 엘리먼트를 추가하거나, 그 속성값 등을 변경하여 정의된 경우, 현재 타입과 그 상위 타입 간에는 수직적인 타입 계층구조가 성립된다.

스키마 브라우저가 스키마 문서에서 정의된 모든 타입과 엘리먼트의 복합 및 타입 계층구조를 그래피컬하게 보여주고, 브라우징 할 수 있는 강력한 기능을 제공하면 스키마 문서를 이해하는데 큰 도움이 된다. 이러한 기능이 제공되면, 사용자는 간단한 마우스 클릭만으로 쉽고 빠르게 그림 1과 같은 타입-엘리먼트 복합 계층구조와 타입 계층구조를 브라우징함으로써 스키마 문서 전체의 구조를 파악할 수 있다. 본 논문에서는 이와 같은 기능을

제공하는 스키마 브라우저를 제안한다.

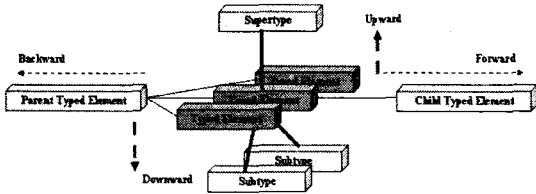


그림 1 타입-엘리먼트 복합 계층구조와 타입 계층구조

제안하는 스키마 브라우저는 그림 1의 스키마 문서 구조를 쉽게 파악하기 위하여 다음 네 가지 방향의 브라우징 기법을 제공한다. Forward 브라우징은 임의의 타입 또는 타입의 바인딩 엘리먼트를 기준으로 자신의 자식 엘리먼트의 정의를 파악하는 방식이다. 반면, backward 브라우징은 임의의 타입 또는 바인딩 엘리먼트를 참조하는, 즉, 자신의 부모 엘리먼트(parent element)를 탐색하는 방법이다. 이와 같이, forward/backward 브라우징 방법을 통하여 타입-엘리먼트 복합 계층구조를 파악할 수 있다. Upward 브라우징은 임의의 타입을 기준으로 상위 타입을 추적하여 상속받은 정의의 구조를 파악하는 방법이고, downward 브라우징은 임의의 타입의 자식 타입들을 탐색하는 방법이다. 따라서, upward/downward 방향으로 브라우징함으로써 타입 계층구조를 이해할 수 있다.

3. XML 스키마 처리를 위한 자료 구조

본 논문에서는 그림 2와 같이 타입-엘리먼트 복합 계층구조와 타입 계층구조를 브라우징 하기 위한 자료 구조를 설계하였다.

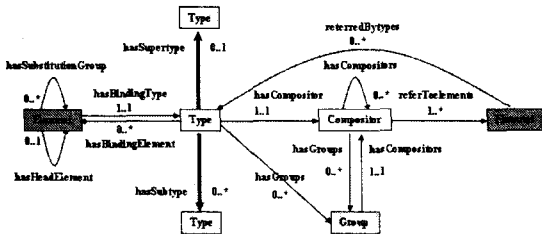


그림 2 스키마 브라우저를 위한 자료 구조

엘리먼트는 하나의 타입에 바인딩되고, 전역으로 정의 되었을 경우, 0개 이상의 멤버 엘리먼트(member element) 및 대표 요소(head element)를 가질 수 있다. 그리고 엘리먼트 자신을 참조하는 부모 타입(parent type)이 0개 이상 정의 될 수 있다.

타입은 정의된 바인딩 엘리먼트(binding element)가 0개 이상 나타날 수 있으며, 최대 1개의 상위 타입(supertype)과 0개 이상의 하위 타입(subtype)으로 타입 계층 구조를 형성 할 수 있다. 단, 리스트(List)와 결합(Union)에 의해 상속 받은 타입은 상위 타입이 1개 이상이 될 수도 있다. 그러나 스키마 브라우저는 전체적으로

트리 구조를 형성하는 것이 기본 규칙이므로, 이러한 경우는 트리의 구조가 깨지게 된다. 그래서 임의의 타입에서 각각의 상위 타입으로 브라우징이 가능하도록 인터페이스를 구현하였다. 또한, 타입은 1개 이상의 컴퍼지터(compositor) 또는 지명 모델 그룹(named model group)을 사용하여 내용 모델(content model)을 정의한다. 컴퍼지터와 지명 모델 그룹은 0개 이상의 컴퍼지터를 가짐으로써 중첩(nesting) 구조를 형성한다[3].

4. XML 스키마 브라우저의 구현

4.1. XML 스키마 브라우저의 전체 구조

스키마 브라우저의 시스템 구성도는 그림 3과 같다. 스키마 브라우저는 스키마 문서의 브라우징을 위한 사용자 인터페이스를 제공한다. 사용자는 navigation control interface, graphical view, textual view를 통해 스키마 문서의 구조를 파악할 수 있는 기능을 제공받는다.

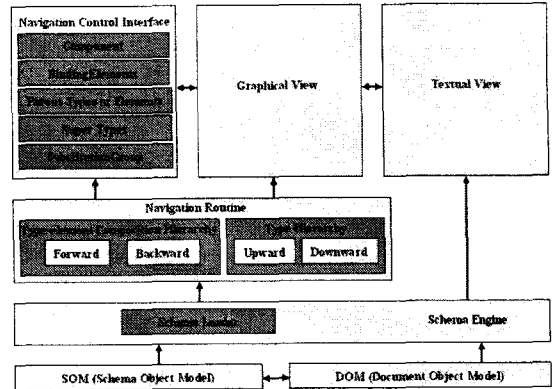


그림 3 스키마 브라우저의 시스템 구성도

Navigation routine은 schema engine을 이용하여 스키마 문서의 구조를 도식화 한 후, graphical view를 통해 사용자에게 보여준다.

Schema engine은 시스템의 최하위 레벨로, 기본적으로 Microsoft 사의 MSXML 라이브러리(MSXML4)를 사용하여 스키마 문서를 파싱(parsing)한 후, 각각의 스키마 컴포넌트 요소 단위로 메모리 상에 로드 한다. 또한 각각의 스키마 컴포넌트 요소 간의 관계를 타입-엘리먼트 복합 계층구조, 타입 계층구조를 위한 내부 자료 구조를 구성한다. 그래서 스키마 문서 내의 스키마 컴포넌트 요소들 간의 관계를 설정하는 작업을 위해 DOM(Document Object Model)과 SOM(Schema Object Model)을 사용한다.

4.2. XML 스키마 브라우저의 클래스 계층 구조

구현한 스키마 브라우저에서 가장 핵심 모듈은 스키마 컴포넌트 요소들 간의 계층 구조를 구축하는 schema engine과 그 계층 구조를 다이어그램으로 구성하여 보여

주는 navigation routine이다. Schema engine은 그림 4와 같은 클래스 계층구조로 구현하였다.

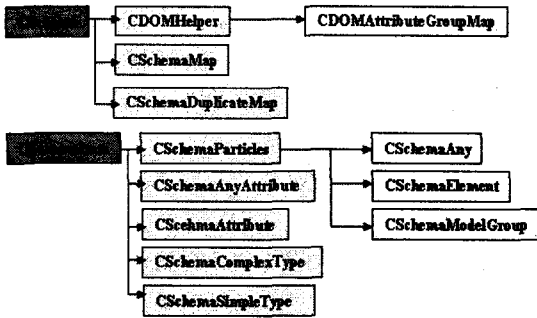


그림 4 Schema engine의 클래스 계층구조

CSchema 클래스는 schema engine의 핵심 클래스로서, 스키마 문서를 구조화하기 위한 루틴이다. 또한 스키마 문서를 처리하는 CDOMHelper 클래스를 구현하였다. 그리고 각 스키마 컴포넌트 요소들간의 관계를 설정할 때 특정 스키마 컴포넌트 요소들을 검색하는 경우가 많기 때문에, 검색 속도의 향상을 위해서 해쉬 테이블(Hash table)을 추가하였다. CSchemaMap 및 CSchemaDuplicateMap은 이를 위한 클래스이다.

Schema engine은 서로 다른 각 스키마 컴포넌트 요소들의 정보를 저장하기 위해서 공통되는 속성을 정의하는 CSchemaItem을 상위 클래스(base class)로 정의하여 이를 기반으로 하위 클래스를 생성한다.

CGraphicalView 클래스는 navigation routine을 위한 클래스로서, 그림 5과 같이 크게 3가지 모듈로 이루어진다. 첫 번째 모듈은 스키마 문서의 구조를 엘리먼트와 바인딩 타입의 값과 속성 값을 정의하는 Display\_Unit을 정의하여 중첩 구조를 형성한다. 두 번째 모듈은 Display\_Unit들의 타입-엘리먼트 복합 계층구조와 타입 계층구조를 graphical view에 나타낼 위치를 정렬하는 과정을 수행한다. 마지막 모듈은 Display\_Unit들을 graphical view의 정해진 위치에 그린다.

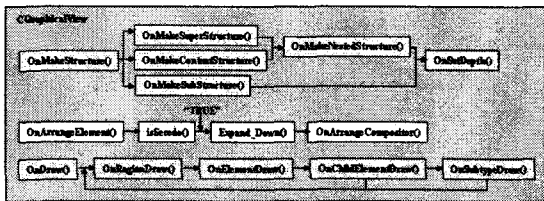


그림 5 Navigation routine을 위한 모듈

5. 사용자 인터페이스

스키마 브라우저의 사용자 인터페이스(이하 UI)는 그림 6과 같다. UI는 크게 navigation control interface, graphical view, textual view, attribute view로 나뉜다.

UI의 왼쪽에 위치하는 navigation control interface는 스키마 문서의 컴포넌트 요소들과의 관계를 리스트로 나

타낸다. Navigation control interface는 네 가지 뷰로 나뉘어진다. Component 뷰는 스키마에서 정의된 모든 컴포넌트 요소(전역(global)/지역(local) 엘리먼트, 복잡/단순 타입, 지명 모델 그룹)을 알파벳 순서대로 보여주고 컴포넌트 요소의 이름을 사용한 검색이 가능하다. BindingElements 뷰는 특정 타입의 바인딩 엘리먼트의 리스트를 제공한다. ParentTypesOrElements 뷰는 특정 엘리먼트에 대해, 그 엘리먼트를 참조하는 모든 타입의 리스트를 제공한다. Supertypes 뷰는 특정 타입의 부모 타입 리스트를 제공한다. 마지막으로 SubstitutionGroup 뷰는 특정 엘리먼트가 대체 그룹을 정의할 경우, 대표 요소와 모든 멤버 엘리먼트의 리스트를 제공한다.

UI의 중앙에 위치하는 Graphical 뷰는 navigation control interface에서 선택된 컴포넌트 요소를 기준으로 타입-엘리먼트 복합 계층구조와 타입 계층구조를 스키마 다이어그램으로 보여준다. 스키마 다이어그램은 그림 파일(.png)로 저장될 수도 있다. Graphical 뷰와 동기화되는 Textual 뷰는 Graphical 뷰에서 선택된 컴포넌트 요소가 하이라이팅(high-lighting)되어 나타난다. 또한, 폴딩(folding) 기능을 제공하여, 매우 길고 복잡한 스키마 문서도 축약한 형태로 볼 수 있다.

Navigation control interface와 graphical 뷰는 아이콘을 사용하여 컴포넌트 요소를 나타낸다. Navigation control interface는 컴포넌트 요소의 종류에 따라 아이콘으로 구분한다. Graphical 뷰는 Display\_Unit 상자(box)의 색깔, 선의 종류, 내부 기호에 따라 컴포넌트 요소의 정의를 표현한다. 그리고 컴퍼지터를 간단한 도형으로 간소화한다. 또한, 선의 굵기와 종류에 따라 계층 구조를 구분하며, 엘리먼트나 타입의 발생 횟수(cardinality)도 나타낸다.

UI의 아래에 위치하는 Attribute 뷰는 특정 요소의 애트리뷰트(attribute) 리스트를 제공한다.

스키마 브라우저는 지금까지 언급한 것 외에 부가적인 기능을 제공한다. 타입 계층구조를 트리 형태로 나타낸다(그림 7 좌). 그리고 선택된 요소에 해당하는 스키마 컴포넌트 요소의 정의만을 Mini 뷰로 볼 수 있다(그림 7 우). 스키마 문서에 포함된 다른 스키마 문서와의 관계(임포트, 인클루드, 재정의(redefine))를 트리 구조로 보여 준다(그림 7 중).

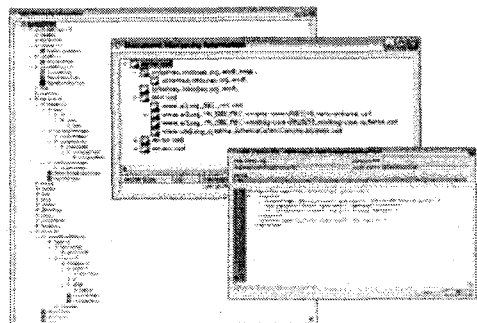


그림 7 TypeHierarchy View(좌), DocumentHierarchy View(중), Mini View(우)

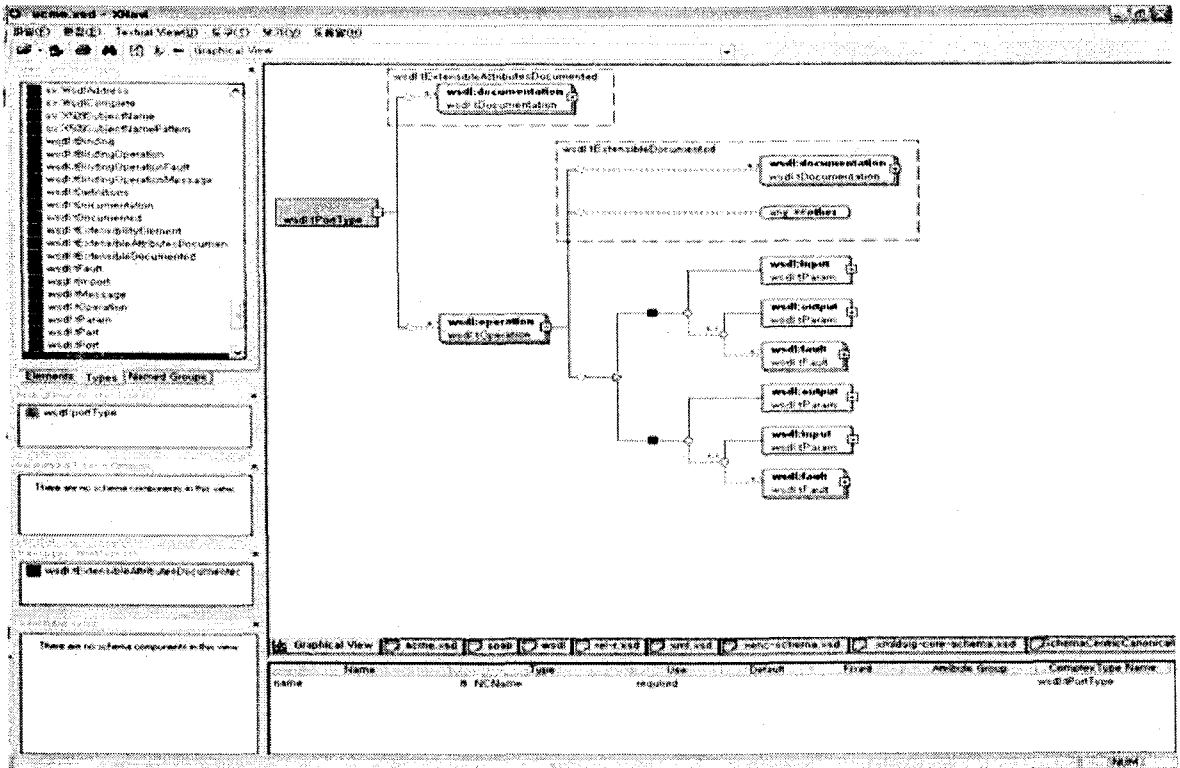


그림 6 스키마 브라우저의 사용자 인터페이스

## 6. 결론

본 논문에서는 복잡한 XML 스키마 문서를 기존의 텍스트 형식의 브라우저의 한계점을 극복하기 위해 타입-엘리먼트 복합 계층구조와 타입 계층구조를 브라우저할 수 있는 그래픽 스키마 브라우저를 제안하였다. 제안한 스키마 브라우저를 사용함으로써, 복잡한 XML Schema의 문법을 모르는 사용자들도 편하게 스키마 문서의 구조를 파악할 수 있다. 앞으로는 사용자 인터페이스의 부가적 기능들을 좀 더 보완하고, 스키마 문서 편집 기능을 추가하여 좀 더 유용한 XML Schema 개발 환경을 구현할 예정이다.

## 참고문헌

- [1] Tim Bray, Jean Paoli, C.M.Sperberg-McQueen, Eve Maler, François Yergeau, "Extensible Markup Language (XML) 1.0", Aug. 2006. (<http://www.w3.org/TR/2006/REC-xml-20060816/>)
- [2] M. Bober, S. R. Quackenbush, C. Seyrat, J. R. Smith, "Text of MPEG-7 Multimedia Content Description Interface — Study of CD — Part 10: Schema Definition", 8-12, December 2003

- [3] Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, "XML Schema Part 1: Structures Second Edition", Oct. 2004. (<http://www.w3.org/TR/xmlschema-1/>)
- [4] G.L.Song and K.Zhang, Visual XML Schemas Based on Reserved Graph Grammars, Proc.Int.Conf.Information Technology: Coding and Computing, Las Vegas, NV, April 5-7, 2004, 687-691
- [5] Altova GmbH, XMLSpy Enterprise Edition User Manual, Internet Document, 2005. (<http://www.altova.com>)