

# ebXML 비즈니스 문서 저작도구(XDocBuilder) 설계 및 구현

박천수\*, 강상승\*, 한우용\*, 손주찬\*  
\*전자통신연구원 비즈니스 지식처리 연구팀  
e-mail : [bettle@etri.re.kr](mailto:bettle@etri.re.kr)

## Design and Implementation of ebXML BD Authoring Tool(XDocBuilder)

Cheon-Shu Park\*, Sang-Seung Kang\*, Woo-Yong Han\*, Joo-Chan Sohn\*  
\*Business Knowledge Research Team, ETRI

### 요 약

ebXML의 핵심 컴포넌트(Core Component)는 범 산업 분야 및 다양한 환경에서 재사용 가능하고 컨텍스트(context)에 영향을 받지 않는 일반적인 빌딩블록(building block)으로, 비즈니스 문서를 구성하기 위한 가장 기본 요소이다. 이러한 핵심 컴포넌트는 비즈니스 컨텍스트에 의해 BIE(Business Information Entity)를 이루게 되며 Syntax binding 을 통해 XML Schema, DTD 등의 형태로 표현된다. 따라서, ebXML 환경에서 사용되는 비즈니스 문서를 포함하여 다양한 종류의 XML Schema, DTD, XML 관련 문서를 쉽게 저작(생성, 검증, 편집)할 수 있는 도구가 필요하다. 본 논문에서는 이러한 요구사항에 적합한 ebXML CC 기반 BD(Business Document) 저작도구로 범용적으로 사용할 수 있는 Schema/DTD 기반 XML 문서 편집기, Schema 문서 편집기, DTD 문서 편집기로 구성된 XDocBuilder를 설계 및 구현 하였다.

### 1. 서론

현재 UN/CEFACT 와 OASIS 에서 진행되고 있는 ebXML (Electronic Business using XML)은 모든 거래 당사자들에게 상호 운용적이며 안전하고 일관성 있는 방법으로 광범위한 e-business 정보 사용이 가능한 개방된 XML 기반의 프레임 워크를 제공하는 것을 목표로 하고 있다[1].

ebXML의 핵심 컴포넌트(Core Component)는 범 산업 분야 및 다양한 환경에서 재사용 가능하고 컨텍스트(context)에 영향을 받지 않는 일반적인 빌딩블록(building block)으로, 비즈니스 문서를 구성하기 위한 가장 기본 요소이다. 이러한 핵심 컴포넌트는 비즈니스 컨텍스트에 의해 BIE(Business Information Entity)를 이루게 되며 Syntax binding 을 통해 XML Schema, DTD, EDI MIG(EDI Message Implementation Guide)등의 형태로 표현된다[2].

따라서, ebXML 환경에서 e-business 를 위해 사용되는 비즈니스 문서를 포함하여 다양한 종류의 XML

Schema, DTD, XML 관련 문서를 쉽게 저작(생성, 검증, 편집)할 수 있는 도구가 필요하다.

이에 본 논문에서는 이러한 요구사항에 적합한 ebXML CC 기반 BD 저작도구로 ebXML Reg/Rep(Registry/Repository)에 접속하여 import/export 된 문서를 쉽게 저작(생성, 검증, 편집)할 수 있는 기능뿐만 아니라, 범용적으로 사용할 수 있는 Schema/DTD 기반 XML 문서 편집기, Schema 문서 편집기, DTD 문서 편집기로 구성된 XDocBuilder를 개발 하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 XDocBuilder의 필요성 및 스키마의 구성요소에 대한 내용을 다루고, 3 장에서는 XDocBuilder 시스템의 전체적인 구조, 스키마의 의미 분석을 위한 세부 처리 모듈, GUI와 XML 데이터간의 효율적인 처리를 위한 모듈에 대해 설명한다. 4 장에서는 시스템 구현에 대하여 살펴보고, 5 장에서는 결론 및 향후 연구 방향에 대해

설명한다.

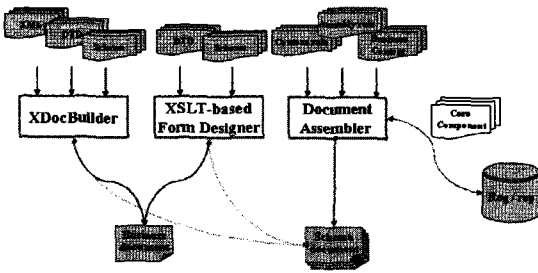
## 2. 배경

### 2.1 ebXML CC 기반 BD 저작 도구

[그림 1]은 ebXML CC 기반 BD 저작도구의 전체적인 시스템의 구성을 나타낸다. 본 시스템은 ebXML Registry/Repository 에 등록된 CC(Core Components)를 기반으로 하여 e-Business 를 수행하기 위해 필요한 BD(Business Documents)를 생성 하는 것을 목적으로 하고 있다. 본 시스템은 크게 Document Assembler, XSLT-based Form Designer, XDocBuilder 로 이루어 진다.

Document Assembler 는 Assembly rule 과 Context rule 을 입력으로 하여 ebXML Reg/Rep 에 저장된 CC 들을 Business context 에 맞게 e-Business 에서 필요한 스키마를 생성하게 된다. 생성된 스키마를 기반으로 XSLT-based Form Designer 에서는 거래할 문서에 대한 양식을 만든 후, 폼을 기반으로 XML 인스턴스를 생성하게 된다.

XDocBuilder 는 Document Assembler 에서 생성된 스키마, XSL-based Form Designer 에서 생성된 XML 인스턴스, DTD 문서를 입력으로 받아 편집, 검증 할 수 있다. 또한 ebXML Reg/Rep 에 접속할 수 있는 클라이언트 API 를 통하여 Rep/Rep 에 import, export 할 수 있는 기능을 제공한다. 본 논문에서는 ebXML CC 기반 BD 저작도구에서 XDocBuilder 에 대한 설계 및 구현한 내용을 설명한다.



<그림 1> ebXML CC 기반 BD 저작도구 구성도

### 2.2 스키마의 구성 요소

XML 스키마는 기본 구성요소와 부가 구성요소로 나눌 수 있다.

먼저, 기본구성요소로는 단순형식정의(simple type definitions), 복잡형식정의(complex type definitions), 요소 선언(element declarations), 속성선언(attribute declarations) 으로 이루어 진다. 단순형식정의는 자식요소를 포함하지 않는 단순 문자열로 string, boolean, decimal, float 등과 같은 원시 데이터형으로 구성된다. 복잡형식정의는 자식으로 다른 요소나 속성을 포함할 수 있는 복합형식이다. 요소 선언은 속성으로 이름(name)과 이름이 가질 수 있는 형태(type)을 가진다. 속성 선언은 속성 이름, 형태, 사용의 선택으로 이루어 진다.

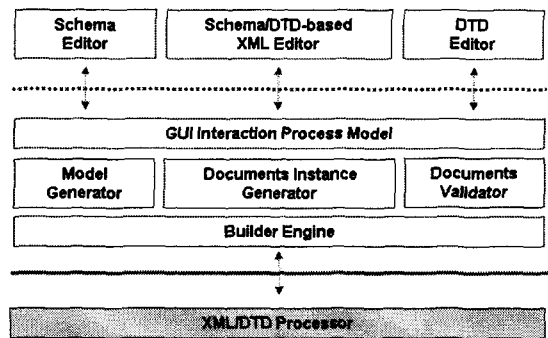
부가 구성요소로는 include, redefine, import, annotation,

compositor(sequence, choice, all), attribute use, wild card(skip, strict, lax), particles(minOccurs, maxOccurs), unique 등으로 이루어 진다. Include 는 동일한 namespace 에 속하거나 namespace 를 가지지 않는 스키마를 가져올 때 사용하고, redefine 은 이미 정의된 스키마에 대한 모델을 재정의 할 때 사용하며, import 는 다른 namespace 에 속해있는 스키마를 가져올 경우 사용한다. Annotation 은 어플리케이션에서 사용할 수 있는 스키마에 대한 설명을 포함한다. Sequence, Choice, All 은 요소에 대한 내용 모델을 제공한다. Attribute use 는 속성대한 제약을 줄 수 있다. Skip, Strict, Lax 는 프로세서가 이 요소 안에 있는 내용의 유효성 검증의 가부를 결정 할 때 사용한다. Particles 는 요소의 출현 횟수를 지정한다. Unique 요소는 한 문서 안에 있는 요소들이 고유한 값을 가지도록 규정할 수 있다[3]~[5].

## 3. 시스템 설계

### 3.1 시스템 구조

[그림 2]은 ebXML CC 기반 BD 저작도구에서 ebXML 에 관련된 XML Schema, DTD, XML 문서들을 저작(생성, 검증, 편집)하기 위한 컴포넌트로 XDocBuilder 시스템 구조를 나타내고 있다. 본 시스템은 크게 XML Schema 를 입력으로 받아 편집하고 새로운 스키마를 생성하는 Schema Editor, XML 문서를 생성, 편집 및 Schema/DTD 기반의 XML 인스턴스를 생성하고 편집 하는 Schema/DTD-based XML Editor, DTD 를 입력으로 받아 편집하고 새로운 DTD 를 생성 하는 DTD Editor 로 구성된다.



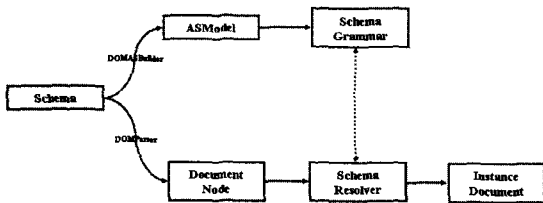
[그림 2] XDocBuilder 구조

내부적인 기능적인 측면에서 각 편집기의 GUI 를 통해서 발생하는 이벤트에 대한 처리를 담당하는 GUI Interaction Process Model 이 있고, 기존의 문서를 편집하거나 새롭게 생성된 문서를 처리하기 위해 메모리 상의 모델을 GUI 에서 Tree/Table 형태로 매핑 하기 위한 Model Generator 가 있다. 또한, Schema/DTD 를 입력으로 받아 새로운 XML 인스턴스 템플릿을 생성하고, 새로운 DTD/Schema/Well-formed XML 문서 생성을 위한 Documents Instance Generator 가 있고, 생성된 문서에 대한 검증을 위한 Documents Validator 가 있다. 마

지막으로 XDocBuilder 내부모듈에는 각 문서 타입에 맞게 XML Processor, DTD processor 를 이용하여 입력 받은 DTD/Schema/XML 문서를 분석하는 모듈, Namespace 를 처리하는 모듈 등을 포함하고 있는 Builder Engine 이 있다.

### 3.2 스키마 처리 모델

[그림 2]는 스키마를 처리하기 위한 모델을 나타낸다. W3C 에서 제공하는 스키마 문법(Schema Grammar) 과 의미를 분석하여 계층적 구조에 대한 관계 정보를 유지하는 Schema Resolver 를 통해 스키마 기반의 XML 문서 인스턴스를 생성한다.

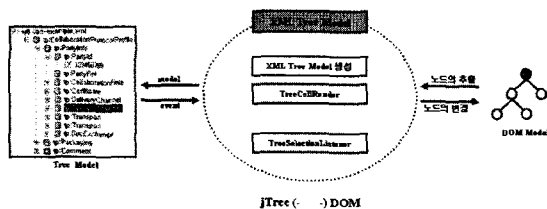


[그림 3] 스키마 처리 모델

입력 받은 스키마를 파싱하여 문서 노드(Document Node)와 ASModel(Abstract Schema Model)을 통한 스키마 문법을 얻어낸다. 스키마 문법에는 하나의 Namespace 내에 선언된 모든 스키마 컴포넌트를 포함하고 있다. 스키마 문법과 문서 노드를 이용하여 Schema Resolve 에서는 각각의 노드에 대한 구조적인 정보를 분석하고, 해석하여 템플릿 문서에 기본적으로 필요한 정보를 제공한다.

### 3.3 사용자 인터페이스(GUI)와 DOM 처리 모델

[그림 4]는 사용자 인터페이스와 DOM 의 처리를 위한 XML Tree 모델을 나타낸다. 본 시스템은 자바 환경에서 구현을 목적으로 하였다. 이에 사용자 인터페이스와 XML 문서에 대한 내부 처리 모델인 DOM 간의 효율적인 인터페이스가 필요하다. DOM 상에서 존재하는 모델을 사용자 인터페이스에서 표현하고 사용자 인터페이스 상에서 발생하는 이벤트를 처리하기 위해서는 여러 가지 방법이 있다.



[그림 4] XML Tree 모델

우선, DOM 문서를 자바 환경의 JTree 에 매핑 하기 위해서는 각 노드를 방문하면서 JTree 에 맞는 XML

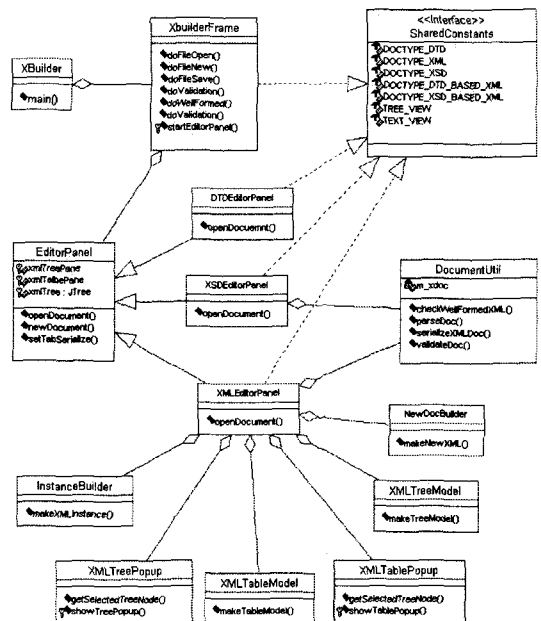
Tree 모델을 생성해 주어야 한다. 생성된 XML Tree 모델을 이용하여 DOM 모델을 JTree 상에 나타내고, 사용자 인터페이스에서 발생한 이벤트 처리는 JTree 상에서의 XPath 표현을 추출하여 XPath 프로세서를 이용하여 처리하는 방법이 있다. 그러나, 이 방법은 이벤트가 발생할 때 마다 XPath 프로세서를 통해 실행한 결과를 가지고 노드의 변경을 해야 하므로 메모리의 낭비를 초래할 수 있다.

이에 본 시스템에서는 사용자 인터페이스와 DOM 이 동시에 사용할 수 있는 XML Tree 모델을 설계하였다. DOM 을 방문하면서 XML Tree 모델을 생성시 DOM 노드 객체를 하나의 JTree 에서의 셀(cell)로 만들어 이벤트 발생시 선택된 JTree 컴포넌트 오브젝트를 XML 노드로 캐스팅 하여 직접 노드의 변경을 할 수 있도록 하여 JTree 와 DOM 을 동기화 하였다.

### 4. 시스템 구현

본 시스템은 자바환경에서 구현하였다. XML Schema, XML, DTD 문서 및 Schema/DTD 기반 XML 문서의 생성 및 편집을 위해 트리(Tree)와 테이블(Table) 구조를 갖는 스윙(Swing) 인터페이스를 사용하였다. 문서의 구조적인 정보와 내부 처리를 위해 Java 기반 XML 파서인 Xerces2 와 DTD 파서를 이용하였고, 자바 개발 환경은 JDK1.4 를 기반으로 개발 하였다.

[그림 5]은 본 시스템에 대한 클래스 다이어그램을 나타낸다. 메인 클래스인 XBuilder 에서 XBuilderInterface 클래스의 open(), new()등의 메소드를 통해 문서의 형태(XML, DTD, Schema)에 맞는 EditorPanel 을 생성하게 된다.



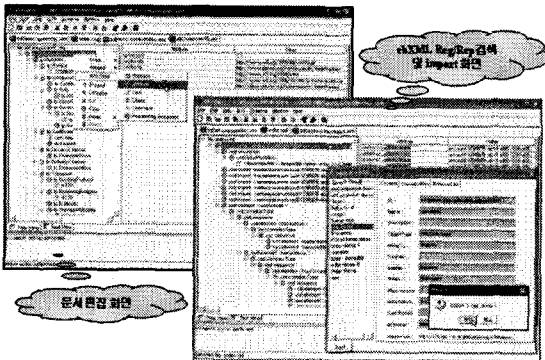
[그림 5] 시스템 클래스 다이어그램

생성된 EditorPanel 에서는 문서의 형태에 맞게 사용자로부터 받은 각각의 행위(action)에 대하여 XMLTreeModel, XMLTable 모델을 생성하여 사용자 인터페이스(GUI)에 문서를 브라우징 한다.

XBuilderFrame 클래스는 사용자 인터페이스에서 요구하는 행위(action)에 대하여 처리와 시스템에 대한 프레임을 관리 하는 역할을 담당한다. EditorPanel 클래스는 추상클래스로 DTDEditorPanel, XMLEditorPanel, XSDEditorPanel 에서 동일하게 사용되는 기능을 추상 메소드로 정의하여 XML, DTD, XML Schema 패널에서 각각의 필요한 사항을 상속 받아 확장하도록 하였다.

XMLEditorPanel 클래스는 입력 받은 문서에 대하여 트리 모델(tree model)과 테이블 모델(table model)의 생성을 담당하는 XMLTreeModel, XMLTableModel 클래스를 근본으로 하여 새로운 인스턴스를 생성하는 역할을 담당하는 InstanceBuilder 클래스와 Well-formed XML 문서 생성을 담당하는 NewDocBuilder 클래스가 있다.

[그림 6]은 ebXML Reg/Rep 에서 문서를 import 하여 편집하는 화면을 나타낸다. 기본적으로 Tree view 와 Text view 로 구성되어 있다. Tree view 에서는 XML 문서에 대한 DOM(Document Object Model) 구조를 Tree 형태로 보여주며, Element, Attribute, PI(Processing Instruction), CData, Comment, Text 에 대한 삽입, 추가, 수정, 삭제 등의 편집을 할 수 있다.

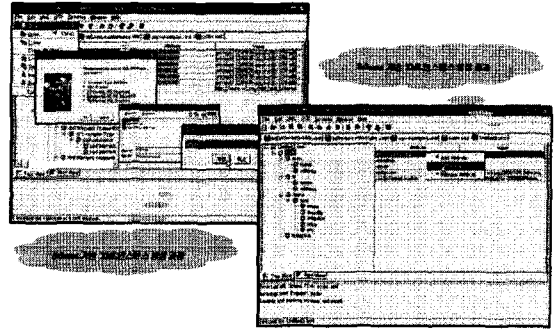


<그림 6> XML 문서 편집 화면

[그림 7]은 스키마 기반 XML 문서 인스턴스 생성 과정과 결과를 나타낸다.

우선, 문서 인스턴스를 생성을 위한 기본 모델인 스키마 파일을 선택한다. 선택된 스키마 파일을 이용하여 최상위 요소인 루트 엘리먼트(Root element)에 올 수 있는 요소들의 목록을 보여준다.

선택된 최 상위 요소가 가질 수 있는 하위 요소들의 정보를 조합하여 필수적인 요소들의 목록들을 구성한다. 즉, 출현횟수가 반드시 한번 이상 나와야 되는 요소들과 속성 값이 required 로 표현된 노드 등에 대한 정보를 이용하여 새로운 형태의 계층적 구조를 갖는 XML 문서 인스턴스를 생성하게 된다.



[그림 7] Schema 기반 XML 인스턴스 생성 화면

## 5. 결론

최근 XML 의 적용분야가 기존의 문서 타입에서 벗어나 전자상거래, e-Business, 데이터 베이스를 비롯한 다양한 애플리케이션에서 처리하는 데이터들을 전송하고 표현해야 되는 필요성을 갖게 되었다. 또한, 현재 UN/CEFACT 와 OASIS 에서 진행되고 있는 ebXML 환경에서 e-business 를 위해 사용되는 문서들을 쉽게 편집, 생성 할 수 있는 도구가 필요하다.

이에 본 논문에서는 ebXML Reg/Rep 에 접속하여 import/export 된 문서를 저작(생성, 검증, 편집)할 수 있는 기능뿐 아니라, 범용적으로 사용할 수 있는 Schema/DTD 기반 XML 문서 편집기, Schema 문서 편집기, DTD 문서 편집기로 구성된 XDocBuilder 를 설계 및 구현 하였다.

## 참고문헌

- [1] ebXML, www.ebxml.org, UN/CEFACT and Oasis.
- [2] ebTWG Core Components, Core Components Technical Specification Part I 1.8, UN/CEFACT, Feb. 2002.
- [3] W3C, XML Schema Part 0:Primer, Recommendation, May. 2001. <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>
- [4] W3C, XML Schema Part 1:Structures, Recommendation, May. 2001. <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>
- [5] W3C, XML Schema Part 2:Datypes, Recommendation, May. 2001. <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>
- [6] Jon Duckett, Stephen Mohr, Oliver Griffin and Francis Norton, "Professional XML Schemas", 2001.
- [7] Lucinda Dykes, Ed Tittel and Ed Tittel, "XML Schemas", 2001.