전표데이터 처리를 위한 XML Schema 설계에 관한 연구

A Study on XML Schema for Slip Data Process

황의철*
광주여자대학교*

Hwang eui-chul*
Kwangju Women's University*

요약

거래내역의 투명성 향상을 위한 기초 자료로 사용되는 전표데이터의 구조를 분석하여 전표 XML Schema를 설계하고, 기업회계의 공통적인 전표처리 기능을 전표데이터베이스와 연동되는 기업 회계와 관련된 다양한 분산 애플리케이션에서 이용할 수 있게 하는, 전표데이터의 표준화와 전표처리 서비스의 정형화 모델을 제시한다.
본 논문의 결과는 전자 상거래에 필요한 회계처리 시스템의 표준화, 웹 서비스에 의한 공통 전표처리기능의 간편한 재사용, 웹 서비스의 접근 용이성을 이용한 기업 간 상호 연동성을 지원하는 표준화의 확산으로 분산 애플리케이션의 효율적인 개발에 기여할 것이다.

Abstract

We analyse the structure of slip data used as basic data for improving the transparency of transactions, and design a slip XML Schema. And then we implement Web Services for slip data, common slip data processing functions of accounting, interacting with the slip database. And we propose a standardization model of slip data structure and slip data processing service in order to make them used to diverse distributed Web applications.
The result of this paper will contribute to the standardization for accounting information system for e-business area, to the simple reuse of common slip data processing functions by Web services, and to the efficient development of distributed applications by grace of spreading standardization for supporting the business-to-business interactivity using easy accessibility of Web services.

I. 서 론

최근 인터넷의 발달로 인하여 웹을 이용한 응용 개발이 매우 활발하게 진척되고 있다. 그러나 지금은 인터넷과 정보통신 기술을 배경으로 한 글로벌 시대임을 감안하면 기업 회계의 기초가 되는 전표 데이터 처리에 대한 웹 환경과 연동이 절실히 요구되고 있으나 아직 이와 관련된 표준화와 정형화에 대한 국내적, 국제적 연구가 부족하다. 최근에 데이터베이스와 초고속정보통신망을 이용한 회계의 평가, 처리 및 전달 기능의 통합으로 인하여 정보의 수집, 처리 및 이용 과정을 신속하게 활용할 수 있는 회계정보 시스템을 구축하고 있고[1], 웹 상에서 데이터 교환의 표준 형식으로 XML이 주목을 받고 있으며 XML은 Schema를 통하여 문서의 구조를 정의할 수 있기 때문에 다양한 형태의 데이터가 XML 문서로 표현할 수 있다. 즉 문서의 구조를 사용자가 원하는 데로 정

의할 수 있으며, 이러한 구조적 유동성은 다양한 형태의 데이터를 XML로 표현할 수 있게 해준다[2].

본 논문에서는 거래내역의 투명성 향상을 위한 기초 자료로 사용되는 전표데이터의 구조를 분석하여 전표 XML Schema를 설계하고, 기업회계의 공통적인 전표처리 기능을 전표데이터베이스와 연동되는 기업 회계와 관련된 다양한 분산 애플리케이션에서 이용할 수 있게 하여, 전표데이터의 표준화와 전표처리 서비스의 정형화 모델을 제시한다.

본 논문의 구성으로 2장에서는 전표데이터를 전표데이터베이스에 저장할 수 있도록 전표데이터베이스 구조를 설계한다. 1)전표데이터 모델 작성, 2)전표데이터베이스 테이블 구조 작성, 3)전표 XML 스키마를 정의하고 전표 XML 스키마 구조도 작성 , 4장에서는 전표 XML 스키마에 근거한 전표 XML 파일을 작성한다. 5장에서는 결론 및 향후 연구방향을 기술한다.

2. 관련 연구

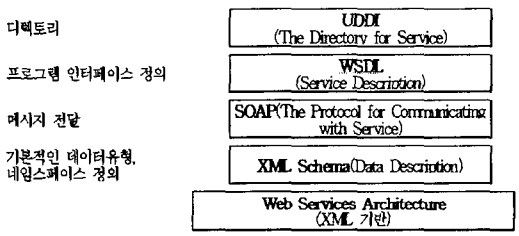
2.1 웹 서비스 구조

전표 데이터의 분석을 통하여 기업의 자금 흐름에 관한 모든 것을 투명하게 파악할 수 있기 때문에 매입, 매출, 투자, 관리상의 효율적 의사 결정을 할 수 있어 기업 생산성을 향상시킬 수 있으나 세금, 보안 등의 문제로 인하여 폐쇄적인 구조를 가지고 있으며 따라서 표준화, 정형화, 공공화에 대한 것은 소극적일 수밖에 없다.

그러나 지금은 인터넷과 정보통신 기술을 배경으로 한 글로벌 시대임을 감안하면 기업 회계의 기초가 되는 전표 데이터 처리에 대한 웹 환경과 연동이 절실히 요구되고 있으나 아직 이와 관련된 표준화와 정형화에 대한 국내적, 국제적 연구가 부족하다.

경영의 투명성과 윤리성이 더욱 강조돼야 할 시점에서 분식회계 사건이 줄을 잇고 있어 심각성을 더하고 있다. 더구나 회계법인의 감사와 당국의 감독이 강화될 경우 회계 대란까지 염려되고 있는 실정이다[3].

현재 W3C가 추진 중인 웹 서비스 표준 규약에서 웹 서비스 구조를 구성하고 있는 기본적인 표준들은 XML(EXtensible Markup Language), SOAP (Simple Object Access Protocol), WSDL(Web Service Description Language), UDDI(Universal Discovery Description and Integration) 등이 있다([그림 1]). XML은 인터넷을 통해 교환되는 데이터 표준 언어로서 오픈 프레임워크인 웹 서비스의 기반 구조를 이루고 있다. XML 스키마(Schema)는 웹 서비스의 기본적인 데이터 유형을 정의하는 역할을 한다.



▶▶ 그림 1. 웹 서비스의 기본구조

XML 스키마는 일종의 데이터 사전으로서 각 개체의 개념을 정의하고 각 데이터에 의미를 부여하여 이질적인 데이터의 상호작용을 가능하게 해준다. SOAP은 분산된 환경의 정보를 교환하는 통신 프로토콜로서 인터넷을 통해 다양한 웹 서비스 사용자가 정보를 교환할 수 있는 통신의 역할을 담당하고 있다. WSDL은 웹 서비스를 정의하는 언어로서 프로그램이나 인터페이스 정의 등 소프트웨어 업체가 웹 서비스를 기술할 때 사용된다. UDDI는 웹 서비스의 디렉토리 서비스를 담당하게 되는데 업체가 자사의 웹 서비스를 온라인 디렉토리에 등록·광고하거나 외부에서 웹 서비스를 검색하는데 사용한다.

[표 1] 전표 XML 스키마(1)

```
<?xml version="1.0" encoding="ks_c_5601-1987" ?>
<xs:schema id="slip_XMLSchema" targetNamespace="http://tempuri.org/slip_XMLSchema.xsd"
elementFormDefault="qualified" xmlns="http://tempuri.org/slip_XMLSchema.xsd"
xmlns:mstns="http://tempuri.org/slip_XMLSchema.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:simpleType name="ID_Type">
                <xs:restriction base="xs:positiveInteger" />
        </xs:simpleType>
        <xs:complexType name="slip_common_Type">
                <xs:sequence>
                        <xs:element name="company_code" type="ID_Type" />
                        <xs:element name="slip_date" type="xs:string" />
                        <xs:element name="slip_no" type="xs:string" />
                        <xs:element name="kind" type="xs:string" />
                        <xs:element name="writer" type="xs:string" />
                        <xs:element name="state_code" type="xs:string" />
                </xs:sequence>
        </xs:complexType>
```

2.2 전표 XML Schema

XML은 자기 기술적(Self-describing)인 데이터 형식으로 설계되어 있기 때문에, 저작자들이 문서의 내용을 기술하는 일단의 요소와 속성들의 이름을 정의하는 것이 가능하게 한다. XML은 이러한 유연성을 허용하므로, 문서를 유용하게 만들기 위해서는 어떤 요소와 속성의 이름들이 사용 가능한지를 부속문서에 정의해 줄 필요가 있다. 또한 각각의 요소와 속성들이 포함할 수 있는 내용에는 어떤 종류가 있어야 하는지도 지시해 주는 것이 필요하다. 이러한 점들을 고려한 스키마 설계로 어떤 XML 문서에 사용된 마크업의 의미를 공유할 수 있게 된다.

다음은 전표 XML Schema로 구성한 예를 설명한다.

▶targetNamespace="http://tempuri.org/slip_XMLSchema.xsd" - 스키마의 큰 개념 중의 하나는 XML 프로세서들이 사용한 문서들을 확인하도록 허락하는 것이다. 그 끝 쪽으로 <schema>엘리먼트는 targetNamespace 라는 새로운 속성을 가진다. targetNamespace 속성은 그 스키마가 겨냥한 네임스페이스를 지정하는 것이다.

이 의미는 만약 XML 프로세서가 문서의 유효성을 확인하고 특정 네임스페이스의 엘리먼트를 체크한다면 스키마의 타깃 네임 스페이스에 기준이 된 어떤 스키마를 체크한다는 것을 알 것이다. 여기서 스키마의 그 타깃 네임스페이스가 http://tempuri. org/slip_XMLSchema.xsd 인 것을 나타내고 있다. 또한 W3C XML 스키마 네임 스페이스 http://www.w3.org/2001/XMLSchema를 만들고 있다.

elementFormDefault="qualified" - local들이 제한적인 것을 요구할 수 있다. 스키마가 엘리먼트 이름들이 문서들을 일치시키는 것에 제한되도록 요구한다[4].

3. 전표 XML Schema 설계

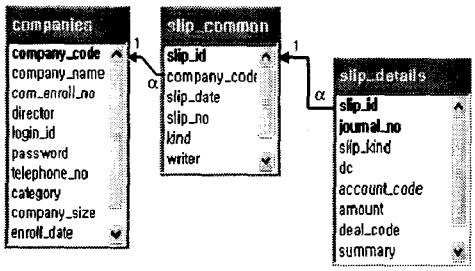
3.1 전표 데이터베이스 모델

전표 데이터베이스 테이블은 실질적인 자료들이 저장, 관리되는 것으로 산재된 자료를 효과적으로 분류하고, 이 자료들을 유용한 데이터로 만드는 것이 전표 데이터베이스의 목적이다. 올바른 테이블의 설계와 관리가 데이터베이스의 효율성을 가름하는 중요한 기준의 하나로 동일한 데이터의 중복을 피하고 다양하고 많은 자료들을 각각 전표특성에 맞게 조정하는 것이 전표테이블을 만드는데 중요한 요소이다.

여러 개의 테이블(전표, 전표내역, 매입매출, 전표확인, 전문가, 회사)이 있는 데이터들을 연관시켜 작업을 한다. 하나의 거대한 테이블보다는 연관된 필드를 중심으로 한 여러 개의 테이블이 좀 더 효과적으로 작동하기 때문이다. 이를 위해 기본 키(회사코드, 전표발생일자, 전표번호)를 설정하게 되는데 그 다음 단계가 바로 테이블과의 관계이다.

• 일대 다 관계

그림 2에서 전표(slip_common)테이블의 한 레코드는 전표내역(slip_ details)의 여러 레코드(분개번호, 전표구분, 대차구분 등)와 일치할 수 있지만, 전표내역 테이블의 한 레코드는 전표에 일치하는 레코드를 한 개만 갖는다.



▶▶ 그림 2. 일대 다 관계의 논리적 모델

그림 2의 논리적 모델을 통해 전표(slip_ common), 회사(company)와 전표내역(slip_details) 실체 사이의 관계를 알아본다. 전표는 회사에서 발행되며, 각 전표별로 분개되는 세부내역인 분개번호(전표 1건에 대한 차변과 대변)나 전표 구분(1.출금, 2.입금, 3.대체차변, 4.대체대변), 대차 구분(1.차변, 2.대변) 등의 관계(relationship)가 있다. 관계차수를 따져 보면, 회사에서는 거래 발생 시마다 전표(인스턴스)가 발행되고, 전표는 여러 전표내역과 관계되므로 일대 다의 관계가 형성된다.

[표 2] 일대 다 관계의 전표 데이터 인스턴스

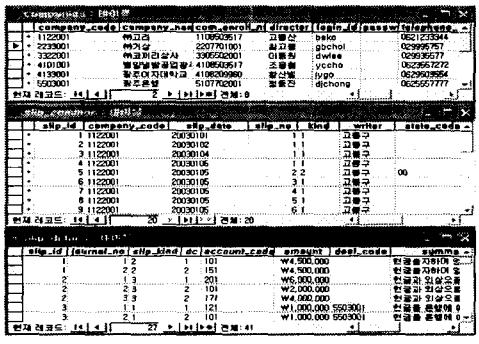
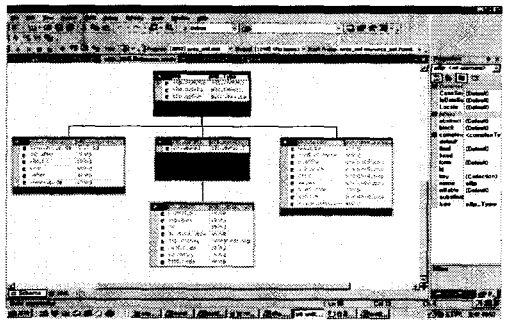


표 2는 일대 다 관계의 전표 데이터 인스턴스로 (a) companies의 회사코드(company_code)가 1122001에 해당하는 (b) slip_common의 company _code는 slip_date, slip_no, kind 등에 따라 9개(slip_id)의 거래가 있음을 보여 일대 다의 관계이다. 또한, (b) slip_common의 slip_id '1'에 대해서도 journal_no의 분개(1.차변, 2.대변), skip_kind(1.출금, 2.입금, 3.대체차변, 4.대체대변), dc(대차구분)가 발생되므로 일대 다의 관계이다.

3.2 전표 XML 스키마 구조도



▶▶ 그림 3. 전표 XML 스키마 구조도

XML 스키마의 중요한 특징 중의 하나는 주요 데이터 타입을 모두 정의해 놓고 있다. 문자열(string), 정수형 (int), 부동소수(float), 더블(Double) 등의 기본 데이터 유형과 integer, decimal과 같은 수학적 데이터 유형 그리고 NMTOKEN과 IDREF 같은 XML XKDLQ 정의를 모두 해 놓았다. XML 스키마의 중요한 장점은 플랫폼에 의존하지 않는다는 것이다. XML 데이터 타입이 지닌값은 하드웨어, 운영체제 및 XML을 처리는 소프트웨어가 무엇이든 간에 일관되게 표현한다. XML 스키마 타입이 있었기에 다양한 시스템간의 상호 운용성을 높여주는 SOAP과 같은 XML 기반 프로토콜의 탄생이 가능하게 된 것이다[5]. 그림 3은 Visual Studio .NET에서 작성된 schema view이다.

```
<?xml version="1.0" encoding="ks_c_5601-1987" ?>
<xs:schema id="slip_XMLSchema" targetNamespace="http://tempuri.org/slip_XMLSchema.xsd" elementFormDefault="qualified"
    xmlns="http://tempuri.org/slip_XMLSchema.xsd" xmlns:mstns="http://tempuri.org/slip_XMLSchema.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"> ①
    <xs:complexType name="slip_common_Type">②
        <xs:sequence>
            <xs:element name="company_code" type="ID_Type" />③
            <xs:element name="slip_date" type="xs:string" />
            <xs:element name="slip_no" type="xs:string" />
            <xs:element name="kind"    type="xs:string" />
            <xs:element name="writer" type="xs:string" minOccurs="0" maxOccurs="1" />④
            <xs:element name="status_code" type="xs:string" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="slip_detail_Type">
        <xs:sequence>
            <xs:element name="journal_no" type="xs:string" />
            <xs:element name="slip_type" type="xs:string" />
            <xs:element name="dc" type="xs:string" />
            <xs:element name="account_code" type="xs:string" />
            <xs:element name="slip_money" type="xs:unsignedLong" />
            <xs:element name="deal_code" type="xs:string" minOccurs="0" maxOccurs="1" />
            <xs:element name="summary" type="xs:string" />
            <xs:element name="field_code" type="xs:string" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="slip_details_Type">
        <xs:sequence>
            <xs:element name="slip_detail" type="slip_detail_Type" minOccurs="2" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="purchase_sales_Type">
        <xs:sequence>
            <xs:element name="tax_type" type="xs:string" />
            <xs:element name="product_name" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <xs:element name="quantity" type="xs:unsignedLong" minOccurs="0" maxOccurs="1" />
            <xs:element name="unit_price" type="xs:unsignedLong" minOccurs="0" maxOccurs="1" />
            <xs:element name="price" type="xs:unsignedLong" />
            <xs:element name="va_tax" type="xs:unsignedLong" minOccurs="0" maxOccurs="1"/>
            <xs:element name="send_date" type="xs:string" />
            <xs:element name="send_no" type="xs:unsignedLong" />
            <xs:element name="image_pathname" type="xs:string" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="slip_Type">
        <xs:sequence>
            <xs:element name="slip_common" type="slip_common_Type" />
            <xs:element name="slip_details" type="slip_details_Type" />
            <xs:element name="slip_option" type="purchase_sales_Type" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
    <xs:element name="slip" type="slip_Type"></xs:element>
</xs:schema>
```

3.3 전표 XML 스키마 설계

XML 스키마를 사용하여 관계형 데이터베이스를 표현하려 할 때는 다음의 사항을 고려해야 한다.

(1) 문서의 범위를 결정한다. - 문서의 목적을 결정하고 그에 따라 문서의 내용을 결정한다.

(2) 형식 정보를 표현한다. - 들어갈 내용의 정보형식들을 결정한 다음, 이 형식을 표현하기 위해 XML 스키마 데이터 형식들을 사용할 것인지를 결정한다.

(3) 탐색 경로(traversal path)를 만들어 준다. - 표현하고자 하는 데이터가 어떤 구조를 가져야 하고 데이터 베이스에서 어떻게 액세스되는지를 분석하여, 문서의 구조가 어떻게 되어야 하는지 결정한다. 그리고 가능한 쉽고 논리적으로 그 경로를 액세스하는데 사용할 수 있도록 그 탐색 경로를 따라 XML로 모델링 한다.

전표 XML 스키마의 기본구조는 다음과 같으며,

```
<slip>
    <slip_common>
            |
    </slip_common>
    <slip_details>
        <slip_detail>
        </slip_detail>
        <slip_detail>
        </slip_detail>
    </slip_details>
</slip>
```

설계된 전표 XML 스키마는 다음과 같다.

스키마를 따르는 문서를 인스턴스 문서라 하며, 이 인스턴스 문서를 위한 XML Schema에 대하여 알아본다. 먼저 여는 Schema 태그에는 XML Schema 권고안에 대한 네임스페이스를 선언하고 있다. XML Schema에 정의된 모든 요소들이 xs: 접두사를 사용하고 있으며, 이 접두사는 schema 루트 요소에 선언되어 있다(①). 다음 줄은 가장 첫 요소인 slip_common_Type 요소를 선언하는 방법을 보여주고 있다. XML은 자기 기술적인 데이터 형식이 되도록 만들어졌으므로 element 요소를 사용하여 요소에 부여할 이름은 name이라는 속성 값에 지정한다. 이 경우엔 루트요소를 slip_common_Type으로 하였다.

sequence 요소는 컴퍼지터(compositor)라고 부르며 complexType 요소의 내부에는 필수적으로 컴퍼지터를 지정해야 한다. complexType은 한 요소가 지닐 수 있는 속성들과 한 요소가 포함할 수 있는 자식 요소들을 정의한다. slip_common_Type 요소는 6개의 자식요소들(company_code, slip_date, slip_no, kind, writer, status_ code)을 포함할 수 있도록 선언되었다(②).

요소 선언은 이름과 형식을 결합시켜 주는 것으로 선언된 요소의 이름은 element 요소가 지닌 name 속성값에 주어진다(③).

기본적으로 XML Schema에 한 요소를 선언하면 그 요소는 필수적으로 오직 한번만 나타나야 되며, 문서 안의 어떤 요소는 선택적으로 나타날 수 있도록 만들어 줄 필요도 있다. DTD(Document Object Model)의 카디널리티 연산자(cardinality operator)인 ?, *, +는 한 요소가 인스턴스 문서 안에서 몇 번 출현할 수 있는지를 지시한다. 이 연산자가 제공하는 기능을 제공하기 위해 XML Schema 요소 선언에서 속성의 형태로 취할 수 있는 두 개의 출현 제한 조건들을 도입했는데 minOccurs와 maxOccurs이다. 이 속성들의 값은 요소가 몇 번이나 나타날 수 있는지를 지시할 수 있다(④). 다음의 예는 전표내역으로 차변과 대변으로 2번 이상 발생하는 경우이다.

```
<xs:element name="slip_detail" type="slip_detail_Type"
 minOccurs="2" />
```

4. 전표 XML 스키마 기반 전표 XML 파일(매입 매출 전표)

매입매출이 발생되는 전표 XML 파일에서

① XML 선언은 xsl 문서, DTD, XML 문서이든 첫 번째 행에 위치하는 전형적인 XML 선언문이다.

② .xsl form 으로 나오게 할 때 추가한다.

③ .xsd 파일보고 규칙 체크한다. 스키마 문법은 tag 규칙만 체크(기본 syntax check)

```
<?xml version="1.0" encoding="ks_c_5601-1987" ?>
<?xml-stylesheet type="text/xsl" href="http://euc.
kwu.ac.kr/cs/websps/slip_form.xsl"?>
<slip xmlns:xsd="http://euc.kwu.ac.kr/ cs/websps/
slip_XMLSchema.xsd">
    <slip_common>
        <company_code>1122001</company_code>
        <slip_date>20020925</slip_date>
        <slip_no>0001</slip_no>
        <kind>1</kind>
        <writer>황의철</writer>
        <status_code>11</status_code>
    </slip_common>
    <slip_details>
        <slip_detail>
            <journal_no>01</journal_no>
            <slip_type>1</slip_type>
            <dc>1</dc>
            <account_code>501</account_code>
```

```
        <slip_money>700000</slip_money>
        <deal_code>1122007</deal_code>
        <summary>비품구입</summary>
    </slip_detail>
    <slip_detail>
        <journal_no>02</journal_no>
        <slip_type>1</slip_type>
        <dc>2</dc>
        <account_code>101</account_code>
        <slip_money>700000</slip_money>
        <deal_code>1122007</deal_code>
        <summary>비품구입</summary>
    </slip_detail>
</slip_details>
</slip>
```

5. 결론 및 향후 연구방향

회계의 공통적인 전표처리 기능을 전표데이터베이스와 연동되는 XML 기반의 웹 서비스로 구현되는 기본 작업으로 전표데이터의 논리적 모델을 통하여 XML 스키마 구조를 설계하는 방법을 연구하였다. XML 스키마가 문서와 데이터베이스들을 정의하고 제한하는데 훨씬 풍부한 기능들을 제공해 줄 수 있다는 사실을 확인하였다. 특히 요소와 속성의 콘텐츠를 구속하는 데 미리 정의된 데이터 타입을 사용할 수 있을 뿐 아니라 심지어는 원하는 값들을 보장하는 자신만의 유도 데이터타입을 만들 수 있다. 이렇게 문서의 유효성을 검증하는 월등한 능력은 제한조건들을 훨씬 강력하게 사용할 수 있게 함으로써, 유효성 검증을 위한 프로그래밍 코드를 작성하는 데 도움이 되는 특징들이 있다[6]. XML 스키마 설계의 연구로 분산환경에서 기업 간의 거래내역, 거래 자료의 공유, 재 사용성 확보, 거래동향 분석 등 효용성있는 처리를 위하여 보다 발전적인 연구를 다각적으로 해야 할 것이다.

▌참고문헌▐

[1] J. Shanmugasundaram, K. Tufte, C. Zhang, D. J. DeWitt, and J. F. Naughton, elational Databases for Querying XML Documents : Limitation and Opportunities,"Proc. of VLDB, Edinburgh, Scotland, pp.302-304, 1999.

[2] Jon Duckett, Professional XML Schemas, pp30-45, 2002, WROX.

[3] "분식회계 실태와 문제점", http://www.hankooki.com/ business

[4] Paul Kulchenko, SOAP *Cross Platform Web Service Development Using XML*, SCOTT SEELY, pp.28-41, 2002.

[5] Scott Short, Bulding XML web services for the microsoft .NET platform, Microsoft Press, pp.137-149, pp.267-268, 2002.

[6] Kevin Williams, professional XML Databases, pp252-263, WROX.