

안전한 메시지교환을 위한 XML 기반의 변환시스템

A XML Based Conversion System for Secure Message Exchange

오 대 균* 홍 창 범** 송 정 길***
Dae-Gyun Oh Chang-Bum Hong Jung-Gil Song

요 약

XML 스키마가 2001년 5월 W3C의 공식 표준안으로 확정되고, XSL과 Xpath, Namespace등의 XML 관련 기술이 발전됨에 따라 XML은 새로운 문서교환의 표준이 되어가고 있다. XML은 다양한 문서의 구조를 표현할 수 있기 때문에 ebXML, RosettaNet, BizTalk들이 e-Business 부분에서 XML을 핵심기술로 사용하고 있다. B2B 전자상거래 시스템에서 업무를 자동화하기 위해서는 표준의 XML 문서가 필요하다. 각 기업들의 내부 문서를 외부기업의 자동화 시스템에서 처리하기 위해서는 문서의 정보손실 없이 외부기업의 문서 형태로 변환하여야 한다. 본 논문에서는 자바 서블릿을 기반으로 XML 문서의 변환하여, 안전한 문서 교환을 제공하는 XML 시스템을 설계 구현하였다.

Abstract

As a XML Schema has been settled to W3C as official standard at May, 2001 and technologies like as XSL, Xpath and NameSpace related with XML has been developing, XML is becoming the standard of the new document exchanging. Due to the ability expressing various types of documentstructure through XML, ebXML, RosettaNet, and BizTalk are using XML as a core technology in the part of e-Business, To make the task automatically in the B2B EC now, a standard XML document is needed. Internal documents in each company should be transformed differently without any loss to work with other companies. In this paper, transforming XML document based on JAVA servlet is considered as realization of a XML system to exchange documents safely.

1. 서 론

최근 HTML과 웹 기술의 발전은 인터넷 대중화에 커다란 공헌을 했다. 이러한 웹 기술은 이제 전자상거래 등의 웹 비즈니스 영역으로 확대됨에 따라 사용자들의 다양한 욕구를 충족시키기에 그 한계점을 드러내기 시작했다. 이러한 기존의 HTML의 한계를 극복하기 위해 XML을 이용함으로써, 데이터 교환뿐만 아니라 데이터 관리와 전송측면에서 전에 없던 유연성을 제공받게 되었다. 현재 XML은 단지 내용과 표현을 구분하기 위하여 사용될 뿐만 아니라, 애플리케이션 사이에 정보의 전달, 원격서비스 호출,

설정 파일의 기술 등 그 활용 영역을 넓혀가고 있다.

지금까지 네트워크를 통한 메시지 교환에는 두 가지의 커다란 기술적 문제점이 존재해왔다. 첫 번째는 메시지 전달 프로토콜에서의 메시지에 대한 도청 및 위조와 같은 공격들로부터의 적절한 정보보호 서비스뿐만 아니라 부인방지와 같은 서비스들의 제공 문제이다. 현재 암호학의 발전과 SSL(Secure Socket Layer)과 같은 프로토콜을 통하여 HTTP에서의 메시지에 대한 기밀성, 무결성, 인증과 같은 정보보호서비스가 제공되고 있으며, 디지털 서명을 이용하여 메시지에 대한 부인봉쇄 서비스를 제공하려는 노력이 이루어지고 있다. 두 번째 문제점은 사용할 표준 메시지 포맷의 부재이다. 각기 독자적인 메시지 형식을 이용하기 때문에 그 메시지를 처리하기 위한 독자적인 시스템을 구축하여 메시지를 교환해야 하는 불편함이 있었지만, XML을 이용하여 플랫폼에 독립적인

* 종신회원 : 한남대학교 컴퓨터공학과 박사과정
dgoh@kier.re.kr

** 준 회원 : 우송대학 강사
cbhong@cc.hannam.ac.kr

*** 종신회원 : 한남대학교 정보통신·멀티미디어공학부 교수
jksong@mail.hannam.ac.kr

호환성과 표준화된 기술을 적용 가능하도록 시스템을 구성하여 다른 시스템과의 연동이 가능하도록 하는 부분에서의 중요성이 증가하고 있다.

XML을 이용하여 메시지 교환이 증가되고 있는 상황에서 XML에 대한 정보보호 기능은 매우 중요하며, 최근에 이러한 XML 메시지뿐만 아니라 일반적으로 네트워크를 통해 전달되는 메시지에 대한 정보보호의 중요성이 강조됨에 따라, 메시지를 대상으로 전자서명과 암호화를 XML 형식으로 생성하여 전자서명과 암호화에 대한 표준화와 효율적이고 체계적인 관리를 위한 XML 전자서명과 XML 암호화에 관한 연구가 활발히 추진되고 있다.

XML을 이용한 XML 전자서명을 생성하기 위해서는 XML 문서의 특성을 고려하여 서명 생성이 이루어져야 한다. 또한 이러한 기업간의 메시지 교환시 현재 XML/EDI, ebXML, RosettaNet, BizTalk 등의 전자상거래를 위한 프레임워크들은 서로 다른 형식의 XML 메시지를 이용하고 있다. 따라서 기업 간의 문서 교환시 이러한 프레임워크의 상이함을 해결하기 위해서는 거래시 원하는 프레임워크에 맞는 메시지 형식으로 변환한 후 XML 전자서명이 적용되어야 한다. 따라서 본 논문에서는 XML 기반으로 기업 간의 물품 구매시 수행되는 모든 프로세스를 온라인 상에서 수행하며, 이때 전송되는 메시지에 대하여 XML 전자서명을 적용하여 효과적이며 신뢰할 수 있는 구매 관리가 이루어질 수 있는 XML 기반의 안전한 전자 구매 시스템을 설계하고 구현하고자 한다.

논문 구성은 다음과 같다. 2장은 관련연구를 설명하고, 3장에서는 변환알고리즘과 변환 규칙, 시스템 설계에 대해 설명하고, 4장에서는 구현결과를 제시하며, 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 XML(Extensible Markup Language)

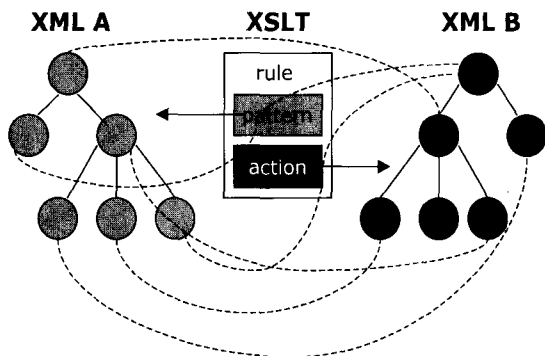
XML은 데이터를 구조적으로 기술하기 위한 마

크업(markup)언어로서 XML 문서는 XML 스펙에 따라 엘리먼트와 애트리뷰트로 구성된다[1]. 엘리먼트는 XML 문서를 구성하는 기본단위이며, 애트리뷰트는 엘리먼트에 부가적인 정보를 기술하기 위하여 사용된다. XML 문서의 구조를 정의하기 위하여 DTD나 XML Schema와 같은 스키마 언어를 이용한다. 이러한 XML과 HTML문서에 대하여 응용프로그램이 접근하는 표준방식을 확립하기 위하여 DOM(Document Object Model)을 이용한다. XML 파서 내부에는 DOM에서 정의해둔 표준인터페이스들을 미리 구현해 놓아서 XML 파서를 이용하여 DOM API를 호출하여 XML 문서를 원하는 방식으로 처리가 가능하도록 한다[2].

2.2 XSL(eXtensible Stylesheet Language)

XSL은 XML 문서의 출력 방법을 정의하기 위한 언어로서 XML 문서 형식으로 기술한다. W3C에 의해 제정된 XSL은 XSLT(XSL Transformations)와 XSL-FO(XSL Formatting Object)로 분리된다[3]. XSLT는 XML 문서를 다른 형식의 문서로 변환시 사용되며, XSL-FO는 문서의 출력 포매팅 내용을 지정할 때 사용된다.

일반 프로그래밍 언어와 달리 XSLT는 순차적인 프로그래밍 언어가 아니라 규칙 기반의 선언적인 프로그래밍 언어로 XSLT에서 선언한 변환에 대한 각각의 규칙들은 서로 독립적이면서 연관성이



(그림 1) XSLT 프로세스

(표 1) XSLT 기본 엘리먼트

엘리먼트	설 명
xsl:stylesheet	XSLT 문서의 루트 노드
xsl:output	XSLT 변환 결과 파일 형식 지정
xsl:template	XSLT 변환 규칙 정의하는 템플릿

있도록 처리된다[4]. 표 1은 이러한 XSLT를 구성하고 있는 기본적인 엘리먼트를 설명하고 있다.

이러한 XSL을 이용함으로써 서로 상이한 포맷의 XML 메시지에 대해서 원하는 포맷의 XML 메시지 형식으로의 변환이 가능하다. 본 논문에서는 XSL을 이용하여 서로 상이한 프레임워크간의 메시지에 대해서 변환시키는 규칙과 알고리즘을 제안한다.

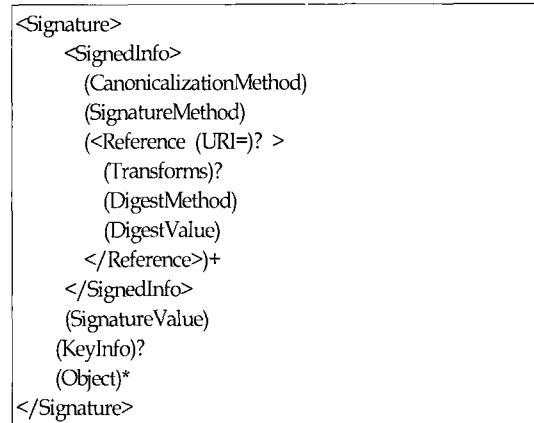
2.3 XML 전자서명

XML 문서는 문서 자체에 구조적인 정보를 포함하고 있기 때문에 XML 문서의 내용에 의미의 부여가 가능해져 XML 문서 단위 또는 일부에 대한 전자서명이 가능할 뿐만 아니라 Manifest를 이용하여 서명시 각각의 키를 이용하여 다량의 문서를 암호화하는데 유용하다. 이러한 특성상 전자서명의 수행에 발생하는 부담을 최소화 할 수 있다. 이러한 XML 문서에 대한 전자서명이 중요해지면서 W3C와 IETF의 XML-Signature 워킹그룹에서는 공동으로 XML 문서 형식을 이용하여 전자서명을 표현하는 방법에 대한 XML-Signature Syntax and Processing, Canonical XML Version 1.0과 같은 XML 전자서명에 대한 기술 표준화를 추진중이다[5].

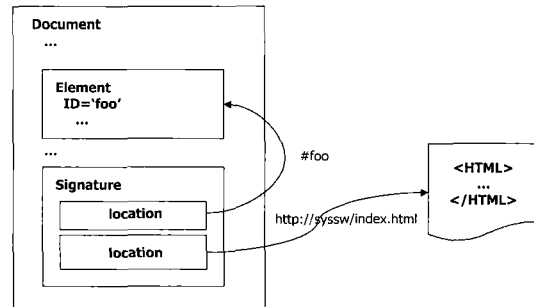
그림 2는 XML 전자서명의 기본 구조로서 XML 형식을 취하고 있으며, 서명될 문서와 XML 전자서명의 포함관계에 따라 다음의 3가지 유형으로 나누어진다.

(1) Detached 전자서명

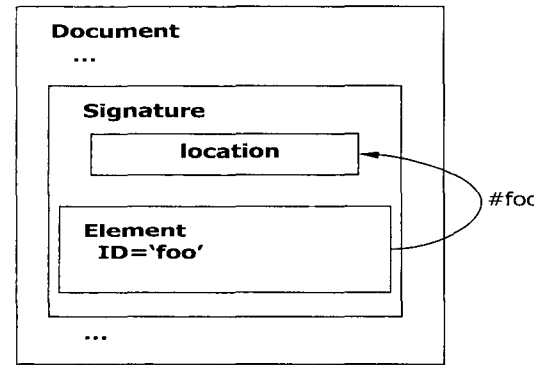
서명될 객체와 서명이 분리된 형태로서, 전자서명 문서와 전송하고자 하는 XML 문서가 분리되어 표현, 전달되어지는 형태이다.



(그림 2) XML Signature 구조



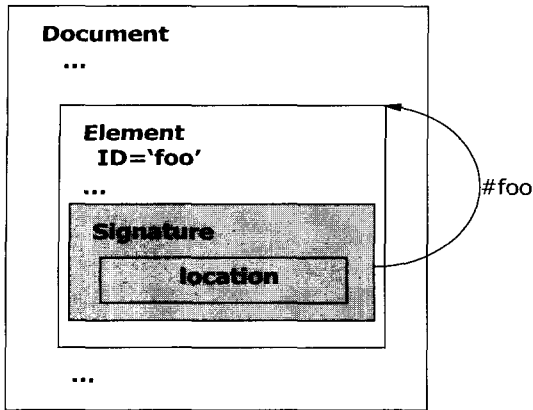
(그림 3) Detached Signature



(그림 4) Enveloping Signature

(2) Enveloping 전자서명

서명될 객체가 서명에 포함되는 형태로서, 서명이 전송 문서의 루트 엘리먼트로 문서 전체가 <Signature>로 시작해서 </Signature>로 끝나게 된다.



(그림 5) Enveloped Signature

(3) Enveloped 전자서명

서명될 객체안에 서명이 포함되는 형태로서, 전송 문서의 자식 엘리먼트로 <Signature> 엘리먼트가 포함되게 된다.

이러한 XML 전자서명의 생성은 다음과 같은 순서로 진행된다.

- ① 서명을 적용할 데이터에 대하여 <Transforms> 엘리먼트에 정의할 알고리즘을 적용한다.
- ② 결과 데이터에 대하여 <Reference> 엘리먼트에 해싱 함수를 적용하여 메시지 축약값을 생성한다.
- ③ 알고리즘과 메시지 축약값을 이용하여 <SignedInfo> 엘리먼트를 생성한다.
- ④ <SignedInfo>에 정의된 암호화 알고리즘을 적용하여 전자서명값을 생성한다.
- ⑤ 전자서명값으로 <SignedValue> 엘리먼트를 생성한다.

생성된 XML 전자서명은 검증과정을 거쳐 실제 안전한 메시지인지를 확인하는 과정을 거치게 된다. 검증과정은 <SignedInfo> 엘리먼트에 포함된 <Reference> 엘리먼트에 포함된 다이제스트를 검증하는 참조검증(Reference Validation)과 <SignedInfo>를 바탕으로 계산된 전자서명을 검증하는 서명검

증(Signature Validation)의 2단계로 이루어진다.

- ① <Reference> 엘리먼트에 명시된 URI가 지정하는 데이터를 얻어온다.
- ② <Reference>에 명시된 해싱 함수를 적용하여 메시지 축약값을 생성한다.
- ③ 검증을 위해 생성된 메시지 축약값과 서명에 포함된 <Reference>에 명시된 메시지 축약값을 비교한다.
- ④ <SignatureValue>도 <KeyInfo>와 <SignedInfo> 정보를 이용하여 확인한다.

3. XML Signature 설계

본 논문에서는 W3C의 XML Signature Core Syntax and Processing을 분석하여 XML 문서에 대한 XML 전자서명을 생성하기 위해 필요한 요구사항과, 알고리즘과 XML 문서의 변환을 위한 프로세스를 기술한다. XML 전자서명은 자바의 표준 API와 자바에서 암호를 사용하기 위한 보안 API인 JCE (Java Cryptography Extension)를 이용하며, XML을 사용하기 위해 오라클의 XML 파서를 이용한다.

3.1 Transform 알고리즘

W3C에서는 XML 문서에 대해서 서명하고자 하는 문서의 일부분 또는 전체를 추출하기 위해서 XSLT, XPath, Enveloped 전자서명 알고리즘을 제시하고 있으며, 각각의 알고리즘과 알고리즘을 표시하는 URI는 표 2와 같다.

(표 2) Transform 알고리즘과 알고리즘 URI

XSLT	OPT	http://www.w3.org/TR/1999/REC-xslt-19991116
XPath	RECOMM	http://www.w3.org/TR/1999/REC-xpath-19991116
Enveloped Signature	REQ	http://www.w3.org/2000/09/xmlsig#enveloped-signature

<Reference> 엘리먼트의 자식엘리먼트인 <Transform> 엘리먼트에 Transform 알고리즘을 기술하여 적용한다. 우선 서명될 객체를 URI를 통해서 얻는다.

```
//Reference를 얻기 위해 URL을 연다.
u = new URL(args[0]);
//URL을 DataInputStream으로 변환하여 읽어들인다.
DataInputStream theREF =
    new DataInputStream(u.openStream());
//읽어들인 XML을 파일로 저장한다.
while((thisLine = theREF.readLine()) != null) {
    fw.write(thisLine);
}
}
```

서명될 객체를 획득한 후에는 Transform 알고리즘을 적용하는데, 여기서는 XSLT 알고리즘을 적용하여 XML 문서를 변환하여 축약 알고리즘을 적용하도록 한다. 전자서명될 원본 객체에 적용할 XSL을 입력받아 XSLT 프로세스를 적용하여 변환한다.

```
//변환된 XML 문서가 저장될 바이트배열의 OutputStream을 생성한다.
ByteArrayOutputStream bout =
    new ByteArrayOutputStream();
//XML 문서를 파싱한다.
DOMParser parser = new DOMParser();
parser.parse(new StringReader(document));
XMLDocument xmldoc = parser.getDocument();
//적용할 XSL문서를 URL을 통해 얻는다.
URL xslURL = url.createURL(stylesheet);
parser.parse(xslURL);
XMLDocument xslDoc = parser.getDocument();
XSLStyleSheet xsl = new XSLStyleSheet(xslDoc, xslURL);
//XML문서에 XSL을 적용한다.
XSLProcessor processor = new XSLProcessor();
processor.processXSL(xsl, xmldoc, bout);
//변환된 XML 문서를 반환한다.
return bout.toString();
```

3.2 Canonicalization 알고리즘

XML문서는 그 특성상 일반적인 사항에 대해서 만족할 수 있도록 정규화 과정을 거쳐야만 하며, 정규화 알고리즘과 알고리즘을 표시하는 URI는 표 3과 같다.

(표 3) Canonicalization 알고리즘과 알고리즘 URI

minimal	RECOMM	http://www.w3.org/2000/09/xmldsig#minimal
Canonical XML with Comments	RECOMM	http://www.w3.org/TR/2000/WD-xml-c14n-20001011#WithComments
Canonical XML	REQ	http://www.w3.org/TR/2000/WD-xml-c14n-20001011

(표 4) Digest 알고리즘과 알고리즘 URI

SHA1	REQ	http://www.w3.org/2000/09/xmldsig#sha1
------	-----	--

3.3 Digest 알고리즘

XML 문서에 대하여 메시지 축약(message digest)을 이용하여 대용량 데이터 집합을 나타내는 식별자를 생성한다. 즉 XML 문서 데이터를 가지고 짧고 축약된 버전의 데이터를 생성하며, 축약 알고리즘과 URI는 표 4와 같다.

```
//다이제스트객체를 생성한다.
MessageDigest md =
    MessageDigest.getInstance("SHA1");
//XML 문서에 다이제스트 알고리즘을 적용한다.
DigestInputStream dis =
    new DigestInputStream(fis, md);
//다이제스트값을 바이트 배열로 저장한다.
byte[] buffer = new byte[8129];
int length;
while ((length = fis.read(buffer)) != -1)
    md.update(buffer, 0, length);
byte[] raw = md.digest();
//바이트배열의 다이제스트값을 Base64로 변환한다.
BASE64Encoder encoder = new BASE64Encoder();
String base64 = encoder.encode(raw);
```

3.4 Signature 알고리즘

XML 문서에 실제 전자서명 생성키를 이용하여 전자서명을 생성한 후 사용된 알고리즘을 해당 엘리먼트에 기술하고, 알고리즘 적용값 또한 XML 엘리먼트로 기술하여 XML 전자서명을 하며, 전자서명 알고리즘과 URI는 표 5와 같다.

(표 5) Signature 알고리즘과 알고리즘 URI

DSAwithSHA1 (DSS)	REQ	http://www.w3.org/2000/09/xmldsig#dsa-sha1
RSAwithSHA1	RECOMM	http://www.w3.org/2000/09/xmldsig#rsa-sha1

```
//지정된 알고리즘을 이용하여 서명객체를 생성한다.
Signature signature = Signature.getInstance("DSA");
//비밀키를 가지고 서명을 초기화 한다.
Signature.initSign(keystore.getPrivateKey(alias, storepass));
//XML문서를 가지고 서명을 갱신한다.
FileInputStream in =
    new FileInputStream(messagefile)
byte[] buffer = new byte[8129];
int length;
while((length = in.read(buffer)) != -1)
    signature.update(buffer, 0, length);
in.close();
```

3.5 구매 요청서 설계

구매 요청서 DTD는 확장성, 반복횟수, 문서 트리의 깊이, 태그 이름 부여 규칙을 고려하여 설계한다. 이는 거래에 참여하는 모든 기업의 요구사항을 만족시키고, 기업이 추가될 경우를 대비하여 확장될 수 있도록 유연하게 설계하기 위해서이다.

구매요청서는 구매 요청서 XML DTD에 정의된 규칙에 따라 HTML 생성폼에 의해서 생성된다. 생성된 XML 형식의 구매 요청서는 데이터베이스에 저장 관리되어 진다.

3.6 구매 요청서 변환

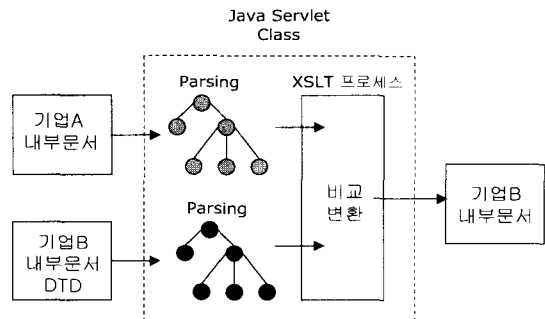
XSLT를 이용하여 전송된 구매요청서 형식이 일치하는지를 검사하여 일치하지 않는다면, 동일한 구조의 XML 문서로 변환작업을 수행한다.

변환된 XML 전자 문서는 자바 서블릿을 통하여 고객의 요구가 있을 때마다 HTML 형식으로 실시간 변환되어 웹 브라우저를 통하여 확인하게 된다.

XSLT 프로세서는 우선 두 문서가 일치하는지를 검증하게 된다. 검증 시에는 그림 8과 같이 XSLT의 <template> 엘리먼트를 이용하여 구매요청서 XML

```
<?xml version="1.0" ?>
<!DOCTYPE PurchaseOrder SYSTEM "po.dtd">
<PurchaseOrder>
  <ShipTo>
    <Name>Hong Gil Dong </Name>
    <Street>Ojung-dong</Street>
    <City>Daejun</City>
  ...
  <Item>
    <ProductNo>Notebook</ProductNo>
    <Quantity>100</Quantity>
    <Price>50000</Price>
  </Item>
</PurchaseOrder>
```

(그림 6) 구매요청서 XML 문서



(그림 7) XML 문서 변환을 위한 구조도

```
<xsl:output method="xml"/>
<xsl:template match="/Order">
  <Order>
    <xsl:apply-templates select="@*|node()" mode="identity"/>
  </Order>
</xsl:template>
<xsl:template match="@*|node()" mode="identity">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()" mode="identity"/>
  </xsl:copy>
</xsl:template>
```

(그림 8) XML 문서 형식 비교

문서의 엘리먼트와 일치하는지를 비교한 후 일치한다면 XSLT의 <copy> 엘리먼트를 이용하여 지정한 노드를 복사한다.

```
protected String style(String document,String stylesheet)
...
{
...
ByteArrayOutputStream bout = new ByteArrayOutputStream();
XSLProcessor processor = new XSLProcessor();
processor.processXSL(xsl, xmldoc, bout);
...
}
```

(그림 9) 변환된 XML 문서의 출력

(표 6) 데이터베이스 스키마

Field Name	Data Type
org	varchar
doc	varchar
addr	varchar
bigo	varchar

두 문서가 일치하지 않을 경우에는 XML 문서를 비교하여 원하는 XML 문서 형식으로 변환한다. 이때, XSL에 정의된 내용이 실제 자바서블릿과 연동되기 위해서는 XSL 문서와 실제 XML 문서를 이용하여 자바의 OutputStream을 통해 동일한 형태의 XML 문서로 변환 출력하게 된다.

변환된 XML 문서를 저장하기 위해 저장시스템을 사용하는데, 현재까지의 XML을 저장하는 방법은 기존의 파일시스템을 이용하는 방법과 관계형 데이터베이스 시스템, XML 전용 데이터베이스 시스템을 이용하는 방법이다.

본 시스템에서는 가장 많이 사용되고 있는 관계형 데이터베이스를 이용하여 XML 문서를 저장한다. 자바의 OutputStream을 통하여 그림 10과 같이 변환된 문서를 저장한다.

4. 구현

본 논문에서 제안하는 XML 문서 교환 시스템을 자바 서블릿을 이용하여 웹 상에서 구현하였다.

XML DOM을 이용하기 위해서 자바용 공개 파서인 오라클의 XML 파서와 아파치 프로젝트의

```
return bout.toString();
}
try
{
PreparedStatement stmt =
connection.prepareStatement("insert into " +
"DOCUMENTS (org, doc, addr, bigo) "
+ "values (?,?,?,true)");
try
{
stmt.setString(1,original);
String document;
document = style(original, stylesheet);
stmt.setString(2,document);
...
}
```

(그림 10) XML문서의 데이터베이스 저장

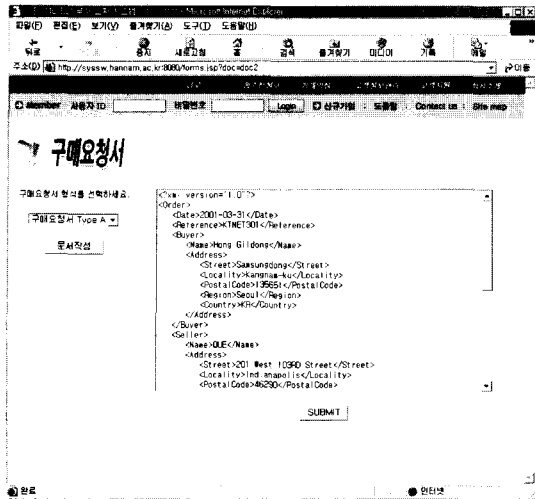
(표 7) 시스템 구성

항 목	내 용
개발언어	JavaServlet - JDK1.3, JSDK, JDBC, JCE
XML Parser	Oracle XML Parser Apache Xerces Java2
Web Server	Apache and Jakarta Tomcat
DB	Oracle 8i for Linux
OS	Linux Kernel 2.4

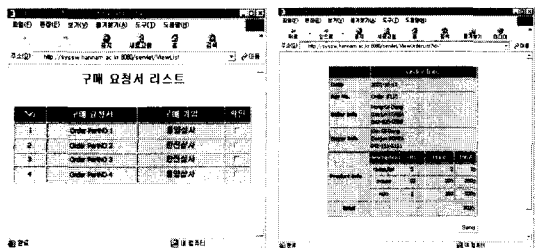
Xerces 파서를 이용하였고, 전자서명을 위하여 SUN의 JCE를 이용하였으며, 데이터베이스는 오라클을 이용하였다.

그림 11은 허브사이트를 방문한 경우 기업의 특성에 따라 원하는 전자문서의 형태에 따른 XML 전자문서의 생성화면으로 HTTP의 Post 방식을 이용하여 문서를 전송한 후 전자서명을 생성하게 된다.

기업 간의 문서가 일치하지 않을 경우 문서의 고유정보가 손실되지 않도록 변환 후 웹을 통하여 문서를 확인하게 된다. 그림 12는 전송된 문서의 리스트를 확인한 후 자세한 내용을 문서에 맞는 XSL을 이용하여 서버 측에서 HTML 형식으로 변환된 결과이다.



(그림 11) 구매요청서 생성 화면



(그림 12) 구매요청서 XML 생성 품

5. 결론 및 향후 연구방향

본 논문에서는 XML의 문서의 변환 기술인 XSLT 기술을 이용하여 네트워크상에서 XML 문서를 교환하고자 할 경우 XSLT를 이용하여 변환이 이루어지도록 프로세서를 설계 구현하여 XML 문서의 다양성으로 인한 문서 교환시의 문제점을 해결하였다. 현재까지 XML 문서의 변환은 XML 문서를 다른 포맷의 문서로 변환하는 부분과 데이터베이스에서 추출된 내용을 XML 형태로 변환하는 부분에 대해서 많은 연구가 이루어지고 있다. 본 논문에서는 XML 문서를 다른 스키마를 사용하는 XML 문서의 형태로 변환하기 위해 연관성이 있는 엘리먼트를 추출하여 웹을 통해 변환이 이루어지도록 하였다. 이때, XML 문서에 대

한 인증을 위하여 XML 전자서명을 이용하여 XML 문서에 전자서명을 추가하여 교환이 이루어지도록 하였다. 따라서 본 논문은 XML 문서를 원하는 형식으로 안전하게 교환할 수 있는 기반 기술이 될 것이다. 향후 연구과제로는 XML 문서 교환을 위한 공개키와 비밀키 생성시 공인인증기관의 연계 부분이 앞으로 추가되어야 하며, 전자서명된 문서에 대해 XML 암호화를 이용한 암호화가 함께 이루어지도록 하는 것이다.

Acknowledgement

이 논문은 2001년도 한남대학교 학술연구조성비 지원에 의하여 연구되었음.

참고 문헌

- [1] Extensible Markup Language(XML) 1.0, <http://www.w3.org/TR/REC-xml>.
- [2] 김형도, B2B 전자상거래@XML, 2000, 배움터.
- [3] XSL Transformations(XSLT) Version 1.0, <http://www.w3.org/TR/xslt>.
- [4] Doug Tidwell, XSLT, 2001, O'Reilly.
- [5] XML-Signature Syntax and Processing, <http://www.w3.org/TR/xmlsig-core/>.
- [6] Westy Rockwell, XML XSLT Java and JSP, 2001, New Riders Publishing.
- [7] Michael Kay, XSLT Programmer's Reference, 2000, Wrox.
- [8] Bill Brogden, Java Developer's guide to E-Commerce with XML and JSP, 2001, SYBEX.
- [9] David Hunter, Beginning XML, 2000, 정보문화사.
- [10] Rational Software Crop., UML for XML Schema Mapping Specification, <http://www.rational.com>.
- [11] XML Schema, <http://www.w3.org/XML/Schema>.
- [13] 강성운, Java WebProgramming, 2001, 가남출판사.
- [14] Brett McLaughlin, Java & XML, 2000, O'Reilly.

◎ 저 자 소개 ◎



오 대 군

1984년 한남대학교 경영학과(경영학사)
1992년 한남대학교 대학원 컴퓨터공학과(공학석사)
1997년 8월~현재 : 한남대학교 대학원 컴퓨터공학과(박사과정)
1979년 10월~현재 : 한국에너지기술원 선임연구원
관심분야 : XML, 실시간 처리시스템, 분산처리시스템
E-mail : dgoh@kier.re.kr



홍 창 범

2000년 청주대학교 생물학과(이학사)
2002년 한남대학교 대학원 컴퓨터공학과(공학석사)
2002년 3월~현재 : 우송대학 강사
관심분야 : XML, 객체지향 모델링 및 방법론(UML), 전자상거래 등
E-mail : cbhong@ce.hannam.ac.kr



송 정 길

1966년 한남대학교 수학과(이학사)
1982년 홍익대학교 대학원 전자계산학과(이학석사)
1988년 중앙대학교 대학원 전자계산학과(이학박사)
1990년~1991년 University of illinois 객원교수
1979년~현재 : 한남대학교 정보통신·멀티미디어 공학부 교수
관심분야 : 멀티미디어 문서처리(XML), 객체지향 모델링 및 방법론(UML), 전자상거래, 분산
시스템 및 실시간 시스템 등
E-mail : jksong@mail.hannam.ac.kr