

지구물리 자료에 대한 XML 기술의 적용

오석훈 · 이덕기 · 남재철 · 양준모

기상연구소 해양기상지진연구소

요 약

인터넷 인프라의 급속한 확충은 각종 자료 및 정보를 신속하고 편리하게 교환하거나 처리할 수 있는 기반을 제공하고 있다. 현재 인터넷 기반 기술로는 HTML이 널리 이용되고 있으나, 다양한 형식의 자료 및 정보를 효율적으로 교환하는 데에는 문제점을 가지고 있다. 최근 이를 극복하기 위해 등장한 기반 기술이 XML(eXtensible Markup Language)이다. XML은 HTML과 비슷한 형식이지만, 내용의 표현양식에 중점을 두는 HTML과 달리, 전달하고자 하는 내용의 분류 및 기술에 있어 훨씬 다양한 방식을 제공하는 사용자 정의형의 인터넷 언어이다. 이미 외국에서는 GIS자료의 교환체제에 XML을 이용하고 있다. 본 연구에서는 지구물리자료의 교환 및 처리에 있어 XML의 장점을 설명하고 이를 이용하여 기상청에 기록된 원거리 지진 자료(teleaseismic data)를 XML 형식으로 기록하는 과정을 논의하고자 한다.

주요어: 인터넷 기술, XML, 지구물리자료 교환, 지진자료의 XML 기록

Seok-Hoon, Oh, Duk-Kee, Lee, Jae-Cheol, Nam and Jun-Mo, Yang, 2001, Application of XML technique to geophysical data. Journal of the Korean Geophysical Society, v. 4, n. 2, p. 133-142

ABSTRACT: The rapid expansion of Internet foundation made it possible to handle various data and information with ease and swift. The HTML is widely used in this circumstance to transfer information via Internet. However, the HTML has limit to describe all aspect of data and information, and the XML is proposed for the replacement. The XML is Internet language that supplies more diverse methods to classify and describe the contents. In this study, some advantages to process geophysical data by XML is discussed and teleaseismic data recorded in KMA were transformed to the XML information.

Key words: Internet, XML, exchange of data, teleaseismic data

(Seok-Hoon, Oh, Duk-Kee, Lee, Jae-Cheol, Nam, and Jun-Mo, Yang, Marine Meteorology & Earthquake Res. Lab., Meteorological Research Institute, 460-18, Shindaebang-Dong, Dongjak-Gu, Seoul 156-720, Korea. email: gimul@metri.re.kr)

1. 서 론

유럽 물리학자들간의 정보교류를 원활히 수행하기 위해 시작된 Web의 언어로 쓰여온 HTML (Hyper Text Markup Language)은 인터넷의 폭발적 관심과 관련 산업의 증진에 큰 역할을 하였다. 이는 비교적 쉬운 문법과 즉석에서 결과를 확인할 수 있는 경량성이 그 원인이라 할 수 있으며, 표현하고자 하는 내용을 비교적 쉽게 꾸미고 다양한 디자인의 적용이 가능하다는 것이 장점이

라 할 수 있다. 그러나 인터넷 Web에 대한 이용이 다양한 분야에 적용되면서 쉽고도 간단한 HTML의 장점이 단점으로 나타나는 사례가 늘어나고 있다.

먼저, HTML은 전달하고자 하는 내용의 표현에는 용이하지만, 내용의 분류나 형식들을 정의하기는 어렵다. 이는 갈수록 다양해지는 각종 자료나 정보들을 Web상에서 분류, 정리하기 어렵게 만들며 이를 위해 재차 데이터베이스나 다른 과정을 거쳐야 한다. 따라서 인터넷 시스템 전체에

부담을 가중하게 되며, 많은 양의 자료를 처리해야 하는 경우 하드웨어 업그레이드는 경제적 비용을 요구한다. 또한 내용 분류의 부재로 인해 Web상의 문서들을 검색하고자 할 경우 원하는 주제의 탐색에 어려움을 겪게되며, ASCII 문자 외의 처리문제, 상이한 운영체제나 시스템 상에서의 자료 교환문제 등이 대두되게 된다.

XML은 이와 같은 문제점을 일거에 해소할 수 있는 진정한 의미의 metadata 용 인터넷 언어이다. XML과 HTML의 차이를 구별짓는 제일 주요한 요소는 바로 XML이 내용의 형식(metadata)을 표현한다는데 있다. 앞서 설명한바와 같이 HTML이 문자의 크기 위치 모양새 등의 외적 표현양식에 기반하여 내용을 보인다면, XML은 자료의 분류, 속성, 포함관계 등 전달하고자 하는 내용의 부가적이지만 본질적 정보를 표현하는 언어이다. 이러한 이유로 CGI와 같은 중간단계를 거치지 않고도 바로 데이터베이스 등에 용이하게 접근할 수 있으며, 운영체제나 사용언어, 시스템 등에 독립적으로 XML내의 내용에 대한 검색 등을 수행할 수 있다. 또한 XML상의 정보에 대한 metadata는 인덱스 역할을 함으로써 인터넷 검색 시, 용이하게 찾고자 하는 주제에 쉽게 접근할 수 있도록

해준다. XML은 이미 인터넷 전자상거래 부분에서 많이 이용되고 있으며, 유럽 지역의 GIS 관련 연구자들은 다양한 형식의 자료 교환을 위해 XML의 도입을 추진하고 있는 상황이다(Houlding 2001, Bosak and Bray 1999).

본 연구에서는 이와 같은 XML의 특징을 구체적으로 논의하고 이를 기상청에 기록되어 있는 원거리 지진자료에 대해 간단히 적용하여 그 효용성을 살펴보고자 한다.

2. XML의 구성요소

XML은 HTML과 마찬가지로 SGML에 속해 있는 조판 언어(markup language)이다(Fig. 1). 즉, 표현하고자 하는 내용의 앞뒤에 조판 부호(tag)를 삽입하여 의미를 부가하는 것이다. HTML의 경우, <H1>, <P> 와 같이 글자의 크기나 행의 삽입을 표현하는 미리 정의된 부호를 삽입하였다. 그러나 XML은 미리 정의된 부호가 있지 않으며, 사용자의 뜻대로 부호를 붙일 수 있어 다양한 의미를 부가할 수 있다. 이와 같은 XML의 특징을 뒷받침하기 위해 필요한 구성요소들로 DOM, DTD, XSL 등이 있다.

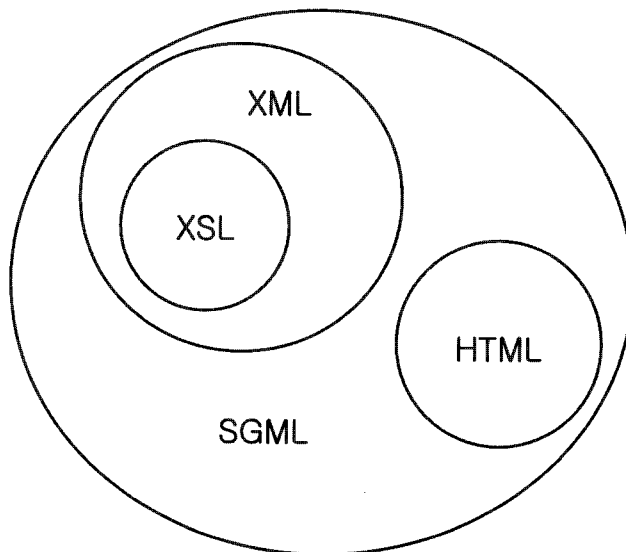


Fig. 1. Relation of inclusion among SGML, HTML, XML and XSL.

2.1. XML의 문법

Fig. 2는 원거리 지진자료의 정보를 XML 형식으로 기록한 예이다. 그림에서 볼 수 있듯이 모든 자료는 의미 있는 부호에 의해 둘러싸여 있으며, 부호만 보더라도 그 의미를 쉽게 이해 할 수 있다. 즉 XML 형식으로 기록되는 정보는 실제의 자료에 대해 지진 발생 일자, 위치, 진원 깊이 등과 같이 우리가 이해할 수 있는 부호로 분류를 함으로써 이후 검색이나 데이터베이스 구축에 매우 용이하게 이용될 수 있다.

문법은 한 필드의 자료를 HTML과 비슷한 방식의 부호가 둘러싸고 있는 양식이며, 레코드의 처음과 끝은 서로 대응을 이루어야 한다(Fig. 2의 Teleseismic의 예 참조). 현재 XML문서는 IE 4.0 이상에서 바로 인식가능하며, Fig. 3은 Fig. 2의 XML 문서를 IE에 띄어 본 결과이다. IE 상에 나타나는 XML 파일은 트리 구조를 갖게되며, 필요에 따라 펼쳐지거나 가려질 수 있어 작성한 내용을 확인 할 수 있다. 이와 같은 XML 문서를

작성할 때는 단순히 에디터에서 만들어도 되지만, MS사에서는 XML Notepad를 무료로 공급하여 작성에 편의를 제공하고 있다(Fig. 4).

2.2. 문서 정의 형식(DTD: Document Type Definition)

Fig. 2나 Fig. 3에서 볼 수 있는 XML 파일은 그 정보를 가공하는 다른 구성요소들에 의해 다양하게 이용될 수 있다. 이를 위해 DOM(Document Object Model)은 C나 Java와 같은 언어 수준에서 XML 정보를 가공할 수 있는 도구를 제공하며, XSL(eXtensible Stylesheet Language)은 XML 문서가 Web 상에서 훌륭하게 표현될 수 있는 양식을 제공한다. Fig. 5는 XML을 구성하는 다양한 요소들을 보여준다.

하지만, 이와 같은 가공을 위해서는 XML 문서에 대해 사용자가 임의대로 정해놓은 부호들의 정의가 공개되어야 한다. 즉, Fig. 2의 경우, Kodiak 지진의 발생 시간은 숫자형이며, 그 부호로는 date가 사용되고, <Earthquake>의 하부구조

```

<Teleseismic>
  <!--Geophysical Data Processing by XML-->
  <Earthquake>
    <title>Kodiak Earthquake</title>
    <info>
      <Description>Felt (VI) at Kodiak, (III) at Anchorage and Seward.
        Also felt at Kenai, King Salmon and Perryville</Description>
      <date>2001 1 10</date>
      <time>16 02 43.4</time>
      <latitude>57.019N</latitude>
      <longitude>153.398W</longitude>
      <depth>33N</depth>
      <Mb>6.1</Mb>
      :
    </USGS>
    <HRV>
      <Mw>7.1</Mw>
      <Me/>
      <ML>6.7 (PMR)</ML>
    </HRV>
  </Fault>
</Earthquake>
</Teleseismic>
  
```

Fig. 2. An example of XML document describing the teleseismic information.

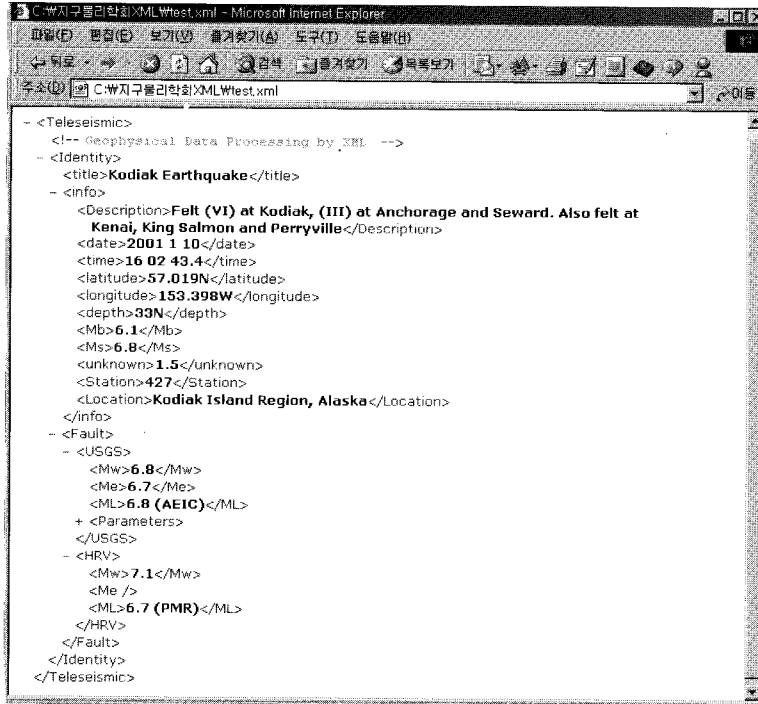


Fig. 3. XML documents loaded in IE 4.0.

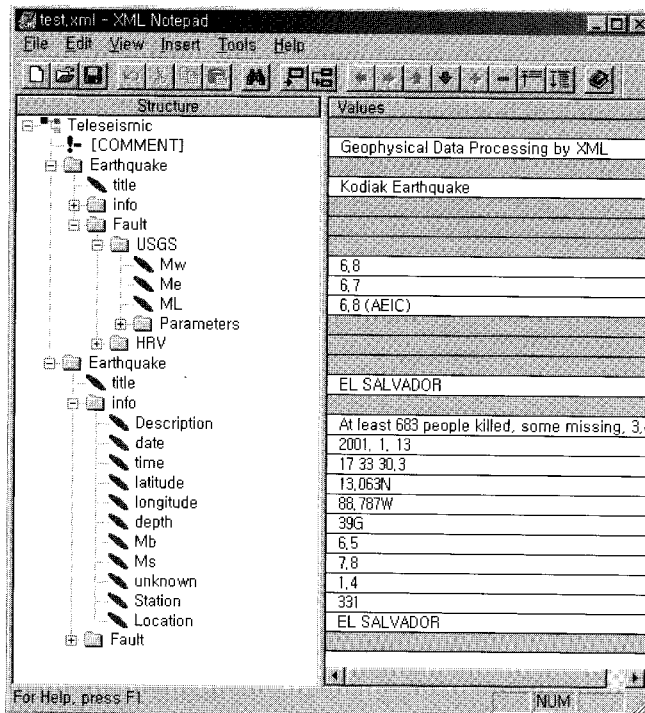


Fig. 4. XML notepad presented by Microsoft which helps to edit the XML document.

인 <info>에 속해 있고 <info>의 하부구조에는 몇 가지 정보가 포함되어 있는지 등에 대한 것이다. Fig. 6은 원거리 지진정보에 대한 DTD 문서의 예를 보인다. 이와 같은 정의가 필요한 이유는

형식을 정의하기 위해서이다. 예를 들어 임의의 원거리 지진정보를 자동으로 XML형식을 이용해 데이터베이스에 추가하는 프로그램을 작성할 경우, 각각의 구성요소에 대한 정의가 필요하기 때

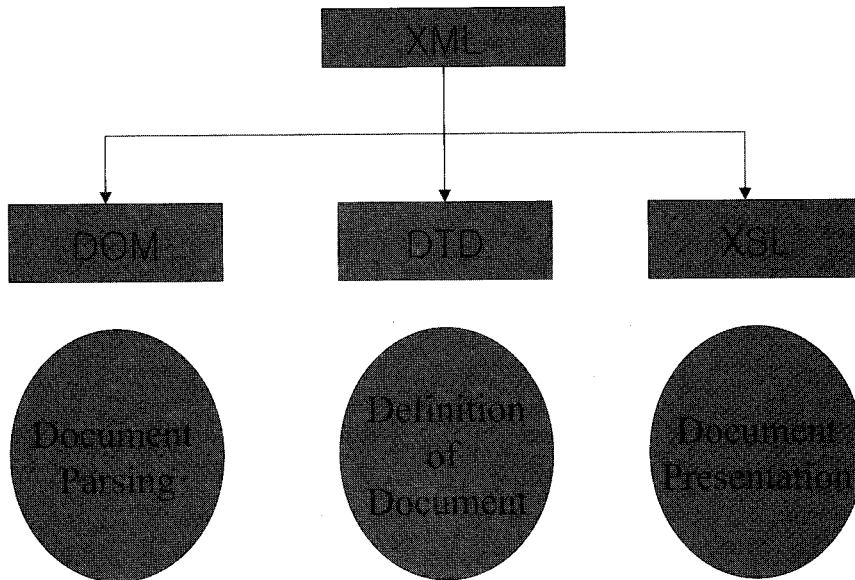


Fig. 5. Various devices to support XML structure.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!ELEMENT Teleseismic ( Earthquake+ ) >
<!ELEMENT Earthquake ( title, info, Fault ) >
<!ELEMENT Fault ( USGS, HRV ) >
<!ELEMENT info ( Description, date, time, latitude, longitude, depth, Mb, Ms,
unknown, Station, Location ) >
<!ELEMENT USGS ( Mw, Me, ML, Parameters ) >
<!ELEMENT Mw ( #PCDATA ) >
<!ELEMENT Parameters ( NP1, NP2, Energy ) >
<!ELEMENT Ms ( #PCDATA ) >
<!ELEMENT date ( #PCDATA ) >
<!ELEMENT HRV ( Mw, Me, ML ) >
<!ELEMENT unknown ( #PCDATA ) >
<!ELEMENT Description ( #PCDATA ) >
<!ELEMENT Strike ( #PCDATA ) >
<!ELEMENT Dip ( #PCDATA ) >
<!ELEMENT depth ( #PCDATA ) >
<!ELEMENT title ( #PCDATA ) >
  
```

Fig. 6. An example of Document Type Definition (DTD) that specifies the format of teleseismic information.

문이다. 또한 차후 설명하게 될, DOM과 XSL에서도 DTD의 형식을 참조하여 검색을 수행하므로 DTD의 작성은 매우 중요하다.

2.3. 문서 객체 모델(DOM: Document Object Model)

인터넷을 이용한 네트워킹이 일반화된 현 시점에서, 네트워크 프로그램의 개발은 필수적이다. 특히 지구물리 자료의 경우, 다양한 헤더와 방대한 자료의 처리로 인해 쉽고 일반적인 네트워크 처리기술의 개발이 당면과제이다. Java 언어는 SUN 마이크로시스템사에서 개발된 객체지향 언어로서, 네트워크 프로그램의 개발에 적합한 언어이다(2000, 권병두 외). 또한 Java는 Java 데이터베이스 연결구조(JDBC)를 이용하여 손쉽게 데이터베이스에 접근할 수 있는 도구도 제공한다.

이와 같은 특징의 Java 언어는 많은 면에서 XML과 유사하다. 즉, 시스템과 운영체제에 독립적이며, ASCII외에 문자 처리가 가능하고 특정 하드웨어나 소프트웨어의 중간자로서 이질적인 두 체제를 연결시킬 수 있는 도구로써 쓰일 수 있다는 것이다. 이런 이유로 Java와 XML은 연동되어 많이 이용되고 있으며, Java를 위한 DOM 역시 많이 쓰이고 있다.

DOM은 XML 문서를 다루기 위한 객체 모델로서 XML 문서에 있는 각 노드를 표현하고 그들 사이의 관계를 이용할 수 있게 한다. 즉, C++, Java 등과 같은 객체지향 언어들을 이용하여 XML 문서의 내용을 분석(parsing), 편집할 수 있도록 객체를 제공하며, 데이터베이스와의 연결을 가능하게 한다. Fig. 7은 Java로 DOM을 이용하여 XML 문서를 다루는 부분의 코드를 보여준다. 즉, DOM에서 제공하는 객체를 로드하여 각종 메소드들을 적용하면, 쉽게 XML 문서의 정보를 가공할 수 있게 된다.

2.4. XML 문서의 표현(XSL)

이미 언급한 바와 같이, XML은 내용의 표현을 위한 도구는 자체적으로 제공하지 않는다. HTML이 부호를 오로지 표현을 위해서는 사용했던 것과 달리, XML에서는 내용의 분류 및 분석에만 이용하기 때문이다. 이는 XML이 단순히 Web 상에서의 내용 전달에 그치는 것이 아니라, 보다 다양한 형식으로 이용될 수 있다는 것을 의미하는 것이지만, 인터넷 이용의 상당 부분을 차지하는 Web에서의 표현을 무시할 수는 없는 실정이다. 이를 위해 제공되는 XSL(eXtensible Stylesheet

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import java.io.File;
import org.w3c.dom.Document;

public class OrderProcessor {
    public static void main (String args[]) {
        File docFile = new File("orders.xml");
        Document doc = null;
        try {
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            doc = db.parse(docFile);
        } catch (Exception e) {
            System.out.print("Problem parsing the file.");
        }
    }
}
```

Fig. 7. An example of document object model (DOM) that render the process of parsing and editing of XML files.

Language)은 XML의 무한한 확장성을 뒷받침하는 진보된 형태의 표현 언어이다. XSL은 기본적으로 XML의 내용을 표현하기 위해 다양한 문서 형식으로서의 전환을 수행한다. 즉, HTML, 그래픽(SVG: Scalable Vector Graphic), PDF 등 XML 파일의 정보를 다양한 형식으로 변환 시켜주는 언어이다(Tidwell 2001).

2.4.1. HTML 변환

Fig. 8은 Fig. 2의 XML 형식으로 기록된 원거리 지진정보를 XSL로 가공하여 HTML 형식으로 변환한 과정과 결과를 보여준다. 즉, XML 문서를 분석하여 각 지진자료의 위치, 날짜 등을 테이블로 작성하며 이를 HTML 형식으로 표출하게 된다. XSL 문서의 대부분은 HTML에서 표현될 테이블을 정의하는데 할애되어 있으며, 테이블에 표출할 자료를 찾는 부분은 <XSL:로 시작되는 부분이며, 테이블 데이터<td>로 쓰여지도록 되어 있다.

2.4.2. SVG(Scalable Vector Graphics)

SVG는 XSL 언어를 이용하여 XML 정보를 그래프로 표현하는 표준 방식으로써, 제목에서 암시하는 바와 같이 벡터 형식을 취하므로써 해상도의 제약을 받지 않고 자유롭게 사용할 수 있는 방식이다. 원칙적으로 2차원 그래픽을 작성할 수 있으며, 텍스트, 다양한 선과 이미지 등을 처리할 수 있다. 또한 도형을 자르거나 변환하고 편집할 수 있으며, 실시간으로 변동되는 XML자료를 상호작용을 통해 그래픽에 반영할 수도 있다.

2.4.3. 기타 형식으로서의 변환

XSL은 위의 형식을 제외하고도 PDF, zip, JPEG 파일 등으로의 변환이 가능하다. 이를 위해서는 일단 XML 문서를 XSL 형식 객체(formatting object: XSL-FO)로 변환한 후, Java나 C++등의 언어에 DOM으로 구현된 객체를 이용하여 목적에 맞게 변경하는 방식을 취하게 된다. 현재 XSL-FO는 웹 서버 프로그램으로 유명한 Apache사에 의해 제안되어 Web의 표준을 관장하는 W3C(World

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/W3-xsl">

<xsl:template match="/">
<html>
<body>
<table border="2" cellpadding="2">
<tr>
<td>Identification</td>
<td>Latitude</td>
<td>Longitude</td>
<td>Date</td>
</tr>
<xsl:for-each select="Telesismic/Earthquake">
<tr>
<td><xsl:value-of select="title" /></td>
<td><xsl:value-of select="info/latitude" /></td>
<td><xsl:value-of select="info/longitude" /></td>
<td><xsl:value-of select="info/date" /></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
    
```

<?xml-stylesheet type="text/xsl" href="test.xsl" ?>

Identification	Latitude	Longitude	Date
Kodiak Earthquake	57.019N	153.398W	2001 1 10
EL SALVADOR	13.063N	88.787W	2001. 1. 13

Fig. 8. XSL routine that converts the XML document to HTML format for describing some element to fit condition.

Wide Web Consortium) 표준 추천을 기다리는 상태이다. 이는 PDF 의 특성을 그대로 이용할 수 있으므로, 세밀한 출력용 작업을 수행할 수 있다. 그림 9는 XSL-FO 규약에 의해 변환된 XML 문서를 PDF 형식으로 변환시켜주는 Java 소스파일의 일부이다.

이 코드는 단순히 FOP 드라이버 객체를 생성하고, 다양한 설정을 수행한 후, PDF 형식으로 변환시키는 과정을 보여준다. XML을 PDF로 변환하는 용도라면 쓴다면 여기에서 제시하고 있는 코드가 충분히 쓰일 수 있으며, XML 문서의 각 요소만 설정을 알맞게 정해주면 될 것이다.

2.5. XML에 의한 자료의 교환

XML 방식에 의한 자료의 교환은 인터넷 상의 사용자 측 컴퓨터(client)를 보다 능률적으로 활용함으로써, 서버 측의 부담을 줄일 수 있다. 즉, 본 정보에 해당하는 자료의 내용은 XML에 실려서 전달이 되므로, 웹으로의 표출이나 각종 포맷으로의 변환, 데이터베이스 작업등은 사용자 측에서 처리할 수 있게 된다. 서버측은 단순히 XSL을 필요에 따라 제공하면 사용자 측의 요구에 맞는 형식으로 변환되므로 네트워크의 부담을 줄일 수 있다(Fig. 10). 또한 XML의 특징인 내용의 분류 형식이 DTD 형식으로 기록될 경우, 보다 강화된 검색 기능을 구현할 수 있다.

2.6. XML과 데이터베이스

XML 문서에서 데이터베이스에 접근하여 필드를 추가, 삭제, 검색하는 등의 방법은 매우 다양하다. 본 연구에서는 최근 새롭게 쓰이고 있는 XSL의 기능 확장(extension element) 방식에 의한 접근해 대해 논의하고자 한다. Extension element란 기존의 XSL이 갖고 있지 못한 기능들을 XSL의 표준을 정의하는 W3C와 같은 공인기관이 아닌 곳(Third Party)에서 개발된 부분을 용이하게 추가하기 정의된 것으로써, XML의 속성(attribute) 부분에 Java 혹은 C++로 작성된 외부 프로그램과 연결할 수 있는 내용(대개 class에 해당함)을 기술할 수 있도록 하는 것이다. Fig. 11은 MySQL이라는 데이터베이스에 접근하기 위해 JDBC 드라이버를 Java 언어에서 연결할 목적으로 작성한 XML 문서의 일부이다.

Fig. 11에서 <dbaccess 이하는 영역은 부호(tag)는 데이터베이스에 접근하는 Java class를 정의하는 요소(element)로 외부 Java 프로그램에 연결된다. 이 부분에서는 데이터베이스로 MySQL을 사용하며 JDBC를 이용하여 로컬 컴퓨터의 test 디렉토리를 접근하고 있다.

이와 같이 XML 문서 내에 정의된 데이터베이스를 연동하기 위해서는 외부 프로그램의 작성이 필요하다. 외부 프로그램은 <dbaccess>라는 요소(element)를 찾아서 그 속성 값으로 들어있는 데

```
public static void buildPDFFile(String foFilename, String pdfFilename)
{
    try
    {
        XMLReader parser =
            (XMLReader) Class.forName("org.apache.xerces.parsers.SAXParser")
                .newInstance();
        Driver driver = new Driver();
        driver.setRenderer("org.apache.fop.render.pdf.PDFRenderer",
            Version.getVersion());
        driver.addElementMapping("org.apache.fop.fo.StandardElementMapping");
        driver.addElementMapping("org.apache.fop.svg.SUGElementMapping");
        driver.addPropertyList("org.apache.fop.fo.StandardPropertyListMapping");
        driver.addPropertyList("org.apache.fop.svg.SUGPropertyListMapping");
        driver.setOutputStream(new FileOutputStream(pdfFilename));
        driver.buildFOTree(parser, new InputSource(foFilename));
        driver.format();
        driver.render();
    }
}
```

Fig. 9. A Java source code to convert the formatting object file into a PDF.

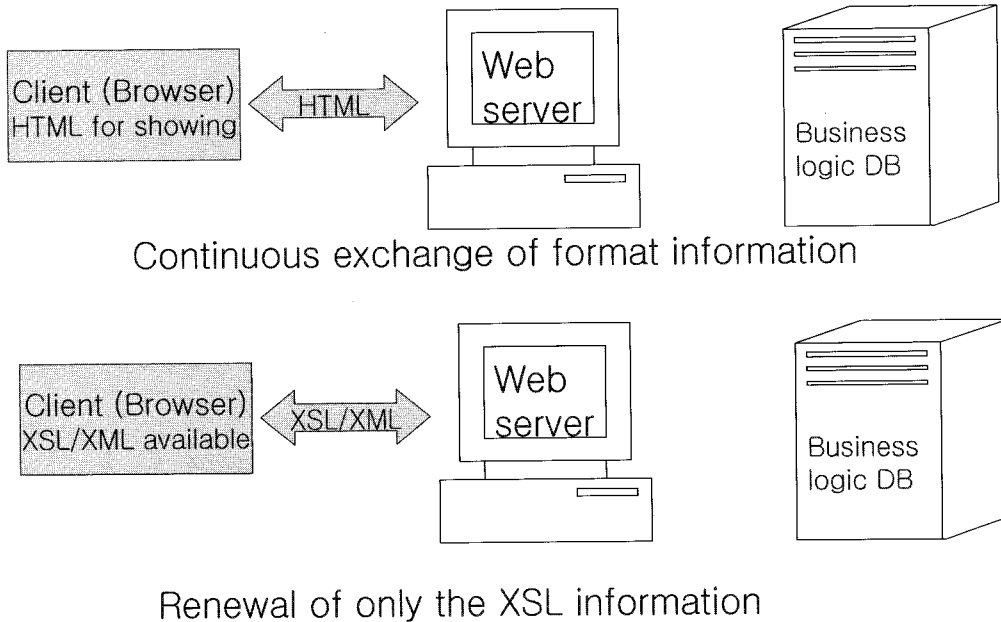


Fig. 10. A change of way for data exchange by Internet Web system. The old way continuously requires communication between server and client for various change of format or display in charge of server side. But the new way effectively distributes the role to both sides and reduces the communication by separation of content and decoration.

```

<?xml version="1.0"?>
<Earthquake>
  <title>Teleseismic Earthquake</title>
  <section>
    <title>yr 2001 Earthquake</title>
    <dbaccess driver="org.gjt.mm.mysql.Driver"
      database="jdbc:mysql://localhost/test"
      tablename="yr2001" where="*" fieldnames='name as "Identification",
        location as "latitude", moment as "moment"' />
  </section>
</report>

```

Fig. 11. An example of XML document implementing the extension element for database access.

이터베이스의 종류와 연결방법에 대해 문의하고, 드라이버를 로드해야 한다. 그 후 XML 문서에서 질의하고자 하는 내용을 분석하여 데이터베이스와 연결하고 처리를 수행한 후, 그 결과를 XSL 루틴과 결합하여 XML문서를 필요시 편집할 수 있도록 해야한다. 이를 위해 Java로 작성되는 외

부 프로그램은 sql, XSL, dom 클래스를 로드해서 각각의 처리 루틴을 작성해야 한다. 이는 Java에서는 매우 간단한 과정이므로 쉽게 작성될 수 있다. 이에 대한 코드는 생략한다.

3. 결 론

XML은 새로운 인터넷 언어로써 다양한 자료의 형식을 정의할 수 있는 강력한 metadata 언어이다. 이는 앞으로 예상되는 네트워크를 통한 방대한 자료의 교환에 있어 엄청난 잠재력을 가질 것으로 예상되며, 다양한 형식의 정의를 통해 자료의 표준을 설정하는데 이용될 수 있다. 이미 유럽 등지에서는 GIS 자료의 교환체계로서 XML을 이용하고 있으며, 그에 알맞은 DTD 등을 정의하고 있다.

XML은 또한 DOM과 XSL을 이용하여 다양한 처리를 수행할 수 있다. 이는 W3C에 의해 규정된 표준 규약으로써, 운영체제나 시스템에 상관없이 쓰일 수 있는 장점이 있다. 기존의 HTML로 처리하기 위해 각 시스템에 맞는 CGI 등을 따로 제작하는 것과는 구분되는 사항이다.

현재 국내에서 XML을 이용하고 있는 분야는 매우 제한적이지만, 전자상거래 분야 등에서 활발히 적용이 진행 중이다. 하지만, XML은 자연과학 분야에서 더욱 효율성을 발휘할 수 있는 부분이므로, 이에 대한 이해의 증진과 신기술에 대한 도입의 노력이 필요하다고 생각된다.

사 사

본 연구는 기상청 기상지진연구개발사업 “지진 예지 및 지진해일 예측기술 개발”의 지원으로 이루어졌으며, 이에 사의를 표합니다.

참고문헌

권병두, 이희순, 오석훈, 정호준, 임형래, 2000, 네트워크에 기반한 중력·자력 자료의 처리기술 개발 연구, 지구물리학회지, 3, 235-244.

Houlding, S. W., 2001, XML - an opportunity for meaningful data standards in the geosciences, Computer & Geosciences, 27, 839-849.

Bosak, J. and Bray, T., 1999, XML and the second-generation Web, Scientific American 5, <http://www.sciam.com/1999/0599issue/0599bosak.html>.

Tidwell, D., 2001, XSLT, O'relly.

2001년 11월 10일 원고접수
 2001년 12월 20일 원고채택