The Journal of Korean
Institute of CALS/EC
Vol 6, No. 2, August 2001

Java를 이용한 XConverter 시스템 설계 및 구현

안경림*, 백혜경*, 임병찬*, 이영교**

The Design and Implementation of XConverter System With Java

K.R. Ahn, H.K. Paik, B.C. Yim, Y.G. Lee

Abstract

AS EC is expanded and activated, many message standards, EDI, XML, UDF, etc are defined and is used at various business part. But the existed translator supports only one mapping translation, for example, EDI-to-UDF, UDF-to-EDI. So many user must be introduced a new translator when new business(message standard) is beginning. To solve this problem, we propose XConverter System as registering mapping information which is able to transform to user requested format using same translator even though it is transferred any formatted message in this paper. Also, We design and implement this system which is capable to integrate with existed legacy system, including DB transformation functionality. The XConverter System supports changing the business message format to the common user used message format. And this system easily migrates with existed legacy system because DB transform module is component. Therefore, The proposed System is more various and flexible than the other translation system that provide just one mapping transformation.

KeyWord : EDI, XML, Translator, Converter, Mapping

* 한국물류정보통신㈜

** LG 전자 정보통신 중앙연구소

1. 서론

기업간, 사용자간 전자상거래가 도입되면서 종이로 처리하던 업무를 점차 전자적으로 메시지를 전송하면서 표준화된 메시지 규격에 대한 요구사항이 도출되었다. 예를 들면, EDI(Electronic Data Interchange)가 대표적인 예라 할 수 있는데, 현재 사용되고 있는 EDI 국제표준으로는 UN/EDIFACT (UN/EDI for Administration, Commerce and Trade), ANSI(American National Standards Institute Accredited Standards Communittee) X12, ODETTE(자동차 산업 표준 : Organization for Data Exchange by Tele-Transmission in Europe), TRADANET(영국과 북아일랜드의 소매산업 표준), SWIFT 등이 존재한다.[1][2][4][5] 사용자가 임의로 정의한 파일을 변환기(Translator)를 통해 표준화된 EDI 문서로 변환하거나 그 반대로 표준 EDI 문서를 사설 포맷으로 변환하는 기능을 수행하였다. 그러나 인터넷 기술이 발달하면서 XML(eXtensible Markup Language)이 도입되어 각 사용자 별로 다양한 메시지 표준들을 정의하여 사용되게 되었다.[3] 전자 거래 분야의 ebXML (Electronic Business XML), 화학 분야의 CML(Chemical Markup Language), 수학 분야의 MathML(Mathematical Markup Language), 웹 서비스의 WIDL(Web Interface Definition Language) 등이 제안되어 표준화 작업 중이거나 현재 사용 중이다.[6][7] 이미 기존 회사에서는 EDI 를 위한 변환기(Translator)와 DB 를 위한 Mapper 등 여러 종류의 변환기를 도입하여 사용하고 있는 실정이다[3]. 그러나 기존의 변환기(Translator)로는 다양한 표준을 지원할 수가 없어, 새로운 시스템을 도입하여야 했으며, 이것은 새로운 메시지 표준이 정의될 때마다 그것에 맞는 변환기가 도입되어야 한다는 가정이 된다. 이를 위해 본 논문에서 제안한 XConverter 시스템은 어떠한 형태의 메시지가 전송되더라도 동일한 변환기를 사용하여 매핑 정보만 등록함으로써, 사용자가 요청한 형태로 변환할 수 있으며, 더 나아가 DB 변환 기능까지 포함하여, 기존 Legacy System(ERP, MIS 등)과도 연동 가능한 시스템으로 설계하였다. 그리고 여러 메시지 형태 중 EDI, XML, DB, HTML 등 일부 형태를 우선 선정하여 구현하였다.

본 논문의 구성을 살펴보면 먼저 제 2 장에서는 변환기에 대한 개념과 XConverter 시스템과 관련된 기술에 대해서 살펴보고, 제 3 장에서는 XConverter 시스템 설계와 이 시스템이 제공할 수 있는 서비스 중 몇 가지를 선정하여 구현한 예를 설명하겠으며, 마지막으로 제 4 장에서는 결론과 향후 해결해야 할 문제에 대해서 언급하겠다.

2. 변환기(Converter) 개념 및 관련 기술

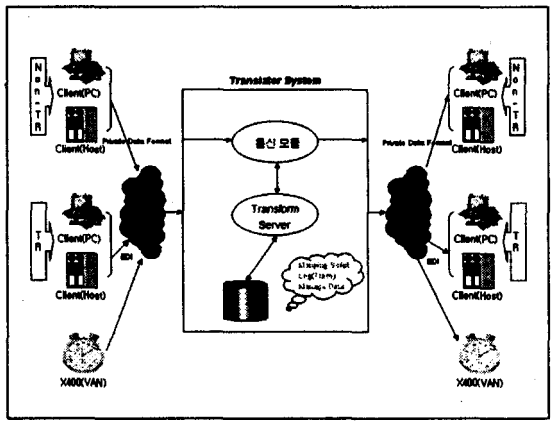
2.1 변환기 (Converter) 개념 및 등장 배경

변환기(Translator)는 표준 형태가 아닌 메시지를 일정한 규칙에 따라 변환하여 표준 형태의 메시지로 변경하거나 그 반대의 행위를 하는 프로세스를 의미한다. 기업간

전자 거래가 도입되면서 EDI가 제안되었으며, 전자적으로 메시지를 주고받게 되면서 각 사용자 별로 사용하던 다양한 문서 표준들을 그대로 적용할 수 없게 되어 공용적인 메시지 표준들이 정의되어 사용하게 되었다. EDI 전자문서 표준으로는 미국 내에서 사용되는 ANSI X12 가 그 외 지역에서는 UN/EDIFACT(우리나라 : KEDIFACT)가 정의되어 사용되고 있다[1][2]. 만약 UN/EDIFACT 표준의 변환기를 가지고 있는 사용자가 ANSI X12 문서 표준의 메시지를 수신하면 변환기를 통해 UN/EDIFACT 문서로 변환하여 내부 시스템에서 사용할 수 있다. 그리고 변환기를 가지고 있지 않는 사용자는 변환기를 보유한 메시지 중계자로부터 사설 포맷으로 변환된 메시지를 수신하여 사용하면 되었다.

2.2 변환기(Converter) 구조

변환기의 구조는 다음 <그림 1>과 같으며, 통신 모듈과 변환 서버, 처리 정보 등으로 구성된다. 통신 모듈은 사용자(변환기 보유/미보유)나 중계업자와 접속하여 표준문서(EDI)나 사설포맷 문서를 주고 받는다. 이 때 사용되는 통신 방식(X25, TCP/IP, 등)은 변환기와 상대 프로세스의 환경에 따라 결정된다.



<그림 1> 변환기 구조

변환 서버는 2 가지 기능을 가지고 있는데, 표준 문서(EDI)로 전송된 메시지를 사설 포맷 문서로 변환하거나 반대로 사설 포맷으로 전송된 문서를 표준 문서(EDI)로 변환하는 기능을 가지고 있다. 또는 하나의 표준 문서(예 : UN/EDIFACT)를 다른 표준 문서(예 : ANSI 12)로 변환하거나 그 반대의 기능을 수행한다. 또한 변환 작업시 표준 문서(EDI)에 대한 검증 작업이 발생하는데, EDI 구조 정보(MIG : Message Implementation Guide) 구조에 의해 행해진다. 마지막으로 처리 정보는 변환 작업을 할 때 사용되는 매핑 정보와 사용자 정보, 문서 정보 등이 저장되어 있는데, 거래 상대방간 정보(사용가능 문서, 버전 등)와 문서 구조 정보(MIG) 등이 포함된다. 변환 작업시 사용되는 처리 정보는 관리자만이 추가, 삭제, 수정 작업을 할 수 있고, 일반 사용자는 권한이 없다.

2.3 관련 기술

2.3.1 자바 (Java)

자바(Java)는 1991 년에 선 마이크로시스템즈(Sun MicroSystems)사에 의해 개발한 객체 지향 프로그램 언어이다. 자바의 문법들은 C 나 C++에서 도입된 것들이 많으며, 복잡하거나 Readability 를 저해하는 요소(C 언어의 Pointer, Memory Control)들을 삭제, 개선시켜, C 나 C++보다는 배우기도 쉬우며 개발하기에도 편리하다. 자바의 특징은 다음과 같다.

- 코드의 단순성
- 객체지향 언어
- 플랫폼 독립

플랫폼에 상관없이 자바 코드를 실행할 수 있는 자바 가상 머신(JVM : Java Virtual Machine) 만 있으면 실행할 수 있다.

- 신뢰성과 안정성

컴파일시에 엄격한 데이터 형을 검사함으로써 프로그램 실행시 발생할 수 있는 비정상적인 상황 등을 미리 막을 수 있다. 시스템의 힙(Heap)이나 스택(Stack) 등의 메모리에 접근할 수 없기 때문에 바이러스로부터 안전하게 보호될 수 있다.

- 높은 성능

쓰레기 수집(Garbage Collection)기능이 있어, 더 이상 사용되지 않거나 메모리가 해제되면 자동적으로 메모리를 수거한다.

- 멀티스레드(Multi-thread)

하나의 프로그램 안에서 여러 작업을 동시에 처리할 수 있는 기능이 있다.

2.3.2 JPython(Java Python)

JPython 은 플랫폼에 독립적이며, 다른 언어(C, Java, Fortran)와도 잘 결합하여, 기존에 구축된 작업환경을 그대로 활용할 수 있다. C 인 경우 SWIG(Simplified Wrapper and Interface Generator)와 같은 좋은 툴이 있어서 소스를 수정하지 않고도 파이썬 확장 모듈로 만들 수 있다. Jpython 이란 순수하게 Java 로 기술된 파이썬 언어로 자바의 각종 패키지를 그대로 활용하며, 자바 바이트코드를 생성해 내며 애플릿(Applet), 서블릿(Servlet) 등도 물론 가능하다. Jpython 의 특징은 다음과 같다[8].

- 단순한 구문
- 동적 형결정
- 내장된 고수준 객체 자료형 (list, tuple, dictionary 등) 지원
- 다양한 표준 라이브러리 지원
- 다양한 써드 파티 유틸리티 지원 (대부분 무료)
- 메모리 자동 관리
- 대형 프로그램을 관리하기 쉬운 모듈 구조

2.3.3 DOM(Document Object Model)

Tree-Based API 로서 XML 문서를 내부 트리 구조로 컴파일하고 응용 프로그램이 그 트리를 내비게이션(Navigation)할 수 있도록 한다. DOM 는 XML 과 HTML 문서를 위해 W3C(World Wide Web Consortium)에서 개발한 Standard Tree-Based API 이다[10].

2.3.4 SAX(Simple API for XML)

SAX 는 XML 파일을 해석하는 프로그램으로서, DOM 에 비해 단순한 구조로서 처리 파일 개수가 많거나 크기가 큰 경우에 적합하다. 그러나 DOM 보다 내용 처리 기능은 부족하다. SAX 는 Event_Driven API 로, 발생할 수 있는 이벤트를 설정하고 그 이벤트가 발생했을 때 해당 행위를 처리한다. XML 문서를 Tree 구조로 컴파일한 후 응용 프로그램이 그 Tree 를 내비게이션(Navigation)할 수 있도록 한다[10].

XML Message 예 :

```
<?xml version="1.0"?>
  <doc>
    <para>Hello,
    world!</para>
  </doc>
```

SAX 적용시

```
start document
start element: doc
start element: para
characters: Hello, world!
end element: para
end element: doc
end document
```

2.3.5 XML(Xpath, XSL)

XML 은 사용자가 직접 태그를 정의할 수 있는 확장이 가능한 메타 언어이며, 이기종 시스템간 호환성과 표준화 기술이다. 특히 현재 전자상거래의 환경이 이질적이고 표준화가 아직 많이 미비한 상태인데 이질적인 시스템간의 데이터 호환을 위해 개발되었기 때문에 전자상 거래 분야에서 매우 활용도가 높을 것이다. XML 은 단순히 문서의 내용을 기술하는 표준 뿐만 아니라 컨텐츠를 포함할 수도 있고 특정 컨텐츠를 표현하는 태그와 속성을 설명하는 DTD (Document Type Definition)을 정의할 수도 있다. XML 의 관련 표준으로는 문법과 관련된 XML, 하이퍼 링크를 지원하는 XLL(XLink, XPointer)와 프리젠테이션 스타일을 프로세싱하여 처리할 수 있는 XSL 등이 존재한다[9][10][11].

XSL 은 기존의 CSS(Cascading Style Sheets)보다 확장성과 기능이 다양하며, 구조화된 XML 문서나 데이터 파일을 어떻게 표현할 것인가를 나타내기 위해 사용된다. XPath 는 XSLT 와 XPointer 사이에 공유된 기능에 공통 구조와 의미를 제공하기 위해 제안되었으며, 동일한 XML 문서나 다른 XML 문서의 특정 지점으로 이동하거나 참조할 수 있다. 이것을 위해 XPath 는 XML 문서를 노드 트리로 모델링하며, Location Path 를 통해 XML 문서 내의 특정 엘리먼트나 속성, PI(Processing Instruction) 등을 지정할 수 있다[9][10][11].

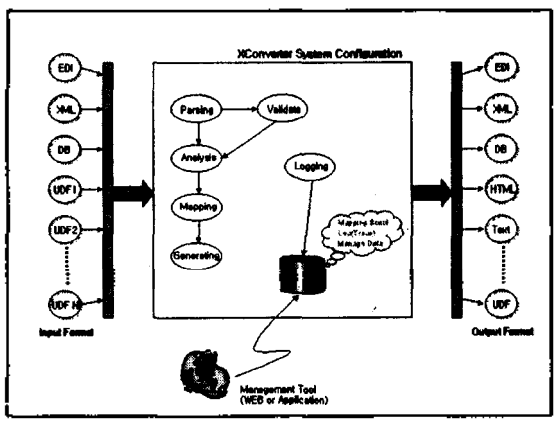
3. XConverter 설계 및 구현

현재 전자상거래 환경하에서 전자적으로 메시지를 교환하기 위해 다양한 표준들이 정의되어 사용되고 있으며, 각 표준(EDI (UN/EDIFACT, ANSI X12), XML, 사설표준(UDF), DB)을 위한 여러 종류의 변환 제품이 사용되고 있다. 그러나 기존 변환기로

는 하나의 표준, 즉 EDI 와 UDF 간 변환, EDI 와 DB, UDF 와 DB 간 변환 등,만을 지원한다. 업무가 추가되거나 확장되었을 때, 서로 다른 표준들끼리 변환하여야 하는 경우에는 별도의 변환기를 도입하여야 하는 실정이다. 그러므로 본 논문에서는 이러한 문제점을 극복하기 위해 각 표준에 대한 매핑 정보만을 등록함으로써 다양한 표준들끼리 상호 변환할 수 있는 Xconverter 시스템을 설계하였다. 그리고 다양한 메시지 표준들중 EDI, XML, DB, HTML, Text 를 선정하여 표준간 변환 서비스를 구현하였다.

3.1 XConverter 시스템 설계

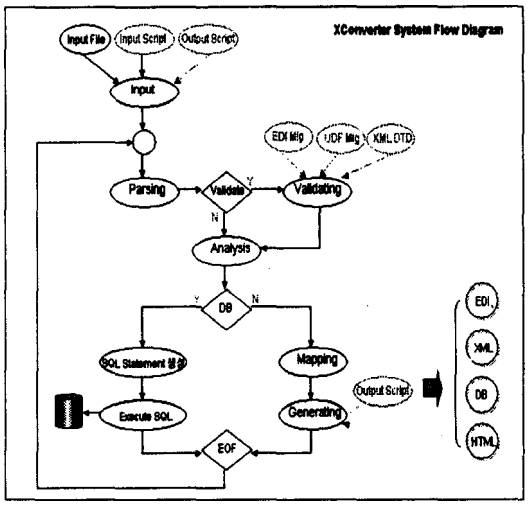
XConverter 시스템은 어떠한 형태의 파일이 전송되어도 로직의 수정없이 등록된 스크립트 정보를 이용하여 사용자가 원하는 형태의 출력물로 변환하는 모듈로서 설계되었으며, XConverter 시스템 구조는 다음 <그림 2>와 같다.



<그림 2> XConverter 시스템 구조

XConverter 시스템의 구조는 파싱(Parsing), 분석(Analysis), 검증(Validate), 매핑, 생성, 로깅, 처리 정보 등으로 구성되어 있다. 먼저 파싱(Parsing) 모듈은 입력된 파일을 입력 스크립트 종류에 따라 데이터를 파싱한다. 등록된 정보에 의해 검증을 요하는 파일 형태일 경우(예를 들면, EDI, XML, UDF 등) 검증 모듈을 호출하여 구문(Syntax), 의미(Semantic) 등의 검증 작업을 수행한다. 분석(Analysis) 모듈은 입력된 메시지의 데이터 중 적합한 값만을 선택하는 모듈로서 동일한 데이터라 할 지라도 결과 파일 형태에 따라 사용 여부가 달라질 수도 있다. 매핑(Mapping) 모듈은 분석 모듈에 의해 결정된 값을 출력 스크립트에 따라 해당 위치(Position)에 매핑한다. 생성(Generate) 모듈은 매핑 모듈에 의해 결정된 정보를 기반으로 하여 결과 파일을 생성한다. 여기서 XConverter 시스템은 기존 시스템과 직접 연동 가능하므로, 만약 결과 파일이나 입력 파일 형태가 DB 일 경우 해당 DB(예를 들면, Oracle, DB2, Informix 등)에 직접 연동하여 DB 값을 추출하거나 결과값을 저장한다. 마지막으로 로깅(Logging) 모듈은 XConverter 내 모듈이 동작할 때 생성되는 로그 정보를 관리하는 모듈로서, 정보는 처리 현황 및 분석 자료로 활용 가능하다. 처리 정보는 사용자 정보와 입력 스크립트, 출력 스크립트, XConverter 시스템 관리 데이터가 저장되어 있으며, 관리자와 사용자가 접속하여 조회할 수 있다. 그러나 스크립트와 관리 정보, 사용자 정보 등과 같은 데이터는 일반 사용자는 조회만 가능하며, 관리자만이 수

정, 삭제할 수 있는 권한이 있다. <그림 3>은 XConverter 시스템의 내부 처리 흐름도를 나타내는 그림으로서 파일과 스크립트를 입력으로 하여 검증과 매핑 작업을 통해 다른 형태의 메시지로 변환되는 과정을 보여주고 있다. 단 출력 형태가 DB 인 경우, 파일로 저장되지 않고 바로 DB 와 연동한다. 즉 XConverter 시스템은 기존 ERP 나 MIS 시스템의 변경없이, 다른 응용 프로그램과 연계하지 않고 DB 시스템과 연동할 수 있다.

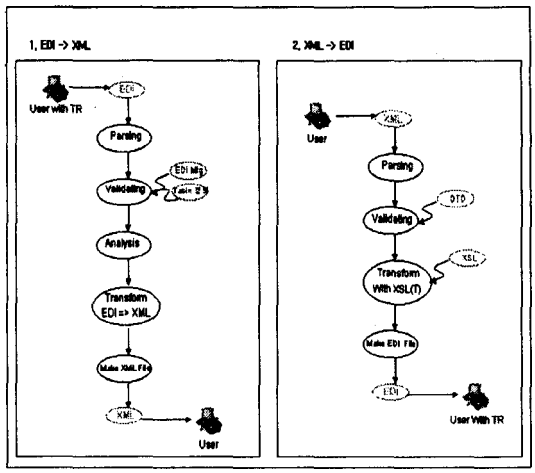


<그림 3> XConverter 시스템 흐름도

3.2 XConverter System 구현

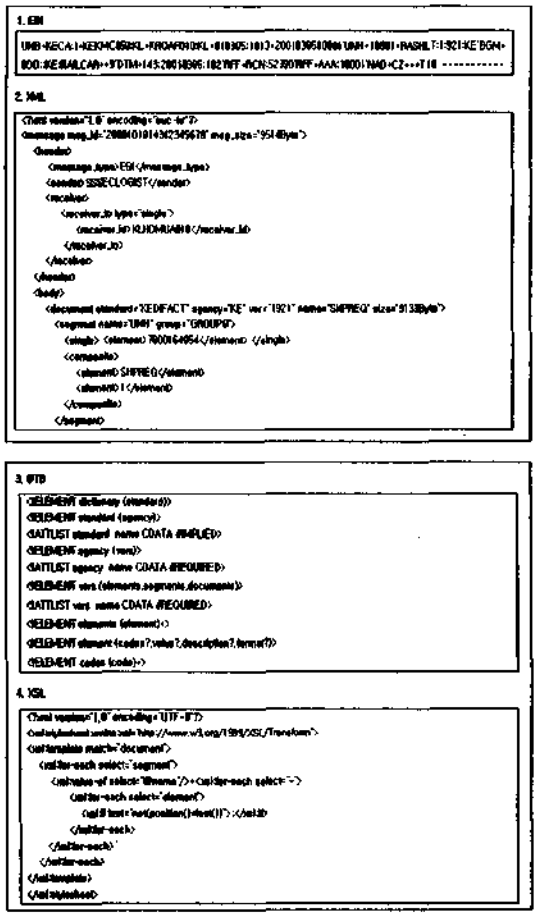
3.2.1 EDI 와 XML 간 변환

EDI 변환기를 보유한 사용자가 EDI 문서를 전송하였을 경우, EDI-To-XML 변환 모듈이 호출된다. 먼저 EDI 문서를 파싱하여 엘리먼트, 세그먼트, 그룹 정보로 분리하여 미리 등록된 EDI 문서 구조 정보(MIG)에 따라 EDI 검증 작업을 수행한다. 오류가 없는 EDI 문서 일 경우 XML DTD 구조에 맞게 값을 셋팅하여 XML 파일로 저장한다. XML 사용자는 XConverter 시스템에 접속하여 원하는 XML 문서를 수신할 수가 있다. (그림 4 의 (1) 참조) 반대로 XML 사용자가 전송한 XML 문서는 파싱한 후, 저장소(Repository) 내 정보를 이용하여 검증작



<그림 4> EDI 와 XML 간 변환

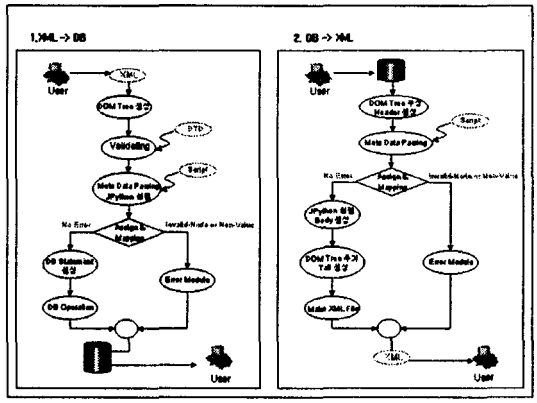
업(Valid-Parser 기능 포함)을 수행한다. 이 작업으로 DTD 구조 Check 뿐만 아니라 Semantic 까지 점검한다. 오류가 없는 XML 문서 일 경우 등록된 EDI 용 XSL 을 적용하여 EDI 문서로 변환한다. 마찬가지로 EDI 사용자도 XConverter 시스템에 접속하여 EDI 문서를 수신할 수 있다.(그림 4 의 (2) 참조) 다음 <그림 4>는 위에서 언급한 사항을 나타내었으며, <그림 5>는 EDI, XML, DTD, XSL 에 대한 예를 보여주고 있다.



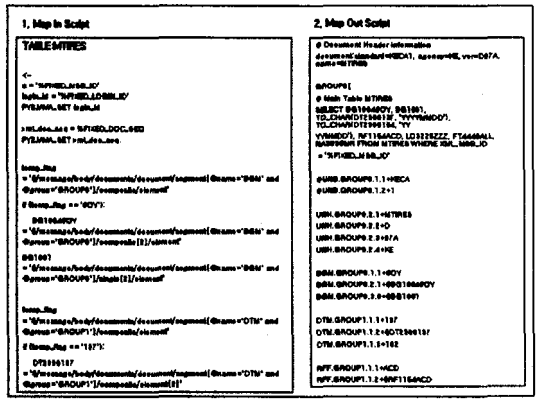
<그림 5> 메시지 예제

3.2.2 XML 과 DB 간의 변환

3.2.1 에서 언급한 EDI<->XML 변환과 유사하게 전송된 XML 문서를 파싱한 후 저장소(Repository) 정보를 이용하여 검증 작업(Valid-Parser 기능 포함)을 수행한다. 오류가 없는 XML 문서 일 경우 해당하는 메타 스크립트 정보에 의해 DB 테이블 항목에 맞게 값을 매핑하여 DB 쿼리문을 생성한다. 이 때 반복 항목일 경우 추가적인 매핑 로직이 필요한데, 본 논문에서는 인터프리터 언어인 JPython 을 이용하였다. Query 문 생성이 완료되면 해당 DB 테이블에 값을 저장한다.(그림 6 의 (1) 참조)



<그림 6> XML 과 DB 간 변환



<그림 7> 스크립트(Script) 예제

반대로 데이터베이스 시스템으로부터 데이터를 추출하여 XML 문서를 생성할 때도 해당 메타 스크립트 정보에 의해 DB 항목 값을 XML 구조에 맞게 매핑하여 DOM Tree 형태로 저장한다. 마찬가지로 별도의 매핑 로직은 JPython 을 사용하였다. 매핑작업이 완료되면 DOM Tree 에 저장된 값을

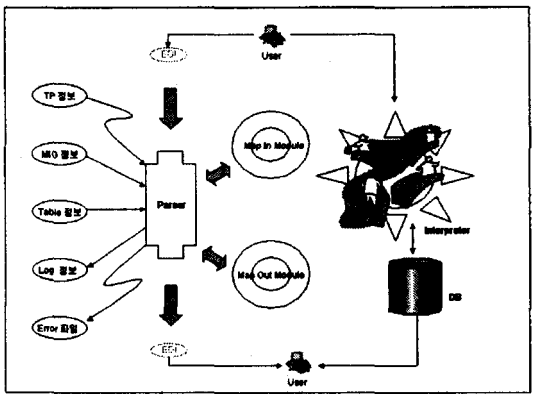
XML 파일로 저장한다. 다음 <그림 6>은 XML-To-DB, DB-To-XML 변환 과정을 보여주고 있다. (<그림 6>의 (2) 참조) <그림 7>은 스크립트 정보의 예제를 보여주고 있다.

3.2.3 EDI 와 DB 간 변환

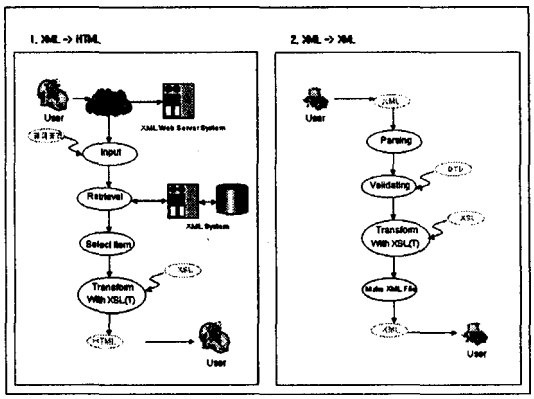
EDI 변환기를 보유한 사용자가 전송한 문서를 DB 시스템에 저장할 경우, EDI-To-DB 변환 모듈이 호출된다. 먼저 EDI 문서를 파싱하여 엘리먼트, 세그먼트, 그룹 정보로 분리하고, 미리 등록된 EDI 문서 구조 정보에 따라 EDI 검증 작업을 수행한다. 오류가 없는 EDI 문서 일 경우 DB 테이블 정보에 의해 각 항목에 맞게 값을 셋팅하여 DB 쿼리문을 생성한다. 변환 모듈은 생성된 DB 쿼리문을 실행하여 DB 시스템에 값을 저장하며, 사용자는 웹을 통해 조회할 수 있다.(<그림 8>의 Map In Module 실행됨) 반대로 DB 로부터 데이터를 추출하여 EDI 문서를 생성할 때도 DB 항목 값을 EDI 구조에 맞게 매핑하여 EDI 문서를 생성한다. (<그림 8>의 Map Out Module 실행됨) 생성된 EDI 문서는 사용자나 타 VAN 등으로 전송될 수 있다. <그림 8>은 EDI <-> DB 변환 모듈의 내부 구조를 보여주고 있다.

3.2.4 XML 과 HTML 변환

변환 시스템이나 독립적인 사용자 프로그램이 없는 사용자 일 경우 유용한 변환으로서, 사용자는 웹 브라우저만으로 메시징 시스템이나 XConverter 시스템에 접속하여 사용하고자 하는 문서를 선택한다. 해당 문서의 검색 조건을 입력하고 검색 버튼을 누르면, 검색 결과가 보여지고, 조회하고자 하는 문서를 선택한다. XConverter 시스템은 XML 로 저장된 문서를 웹 브라우저 상에서 조회할 수 있도록 해당하는 XML 문서에 등록된 HTML 용 XSL 을 적용하여 XSLT 프로세스를 통해 HTML 로 변환한다. 그러면 사용자는 웹 브라우저로 조회할 수 있으며, 검색된 문서를 자신의 시스템에 저장하길 원하는 사용자는 별도의 문서 수신작업을 요청하면, EDI 나 XML 형태로도 저장할 수



<그림 8> EDI 와 DB 간 변환



<그림 9> XML 과 HTML, XML 과 XML 간 변환

있다(<그림 9>의 (1) 참조). <그림 10>은 HTML 용 XSL 의 예제를 나타내고 있다.

3.2.5 XML 과 XML 간 변환

앞에서 언급한 바와 같이 XML 은 다양한 표준(DTD)이 존재할 수 있다. 그러므로, 다른 형태의 XML 문서가 전송되었을 경우 XConverter 시스템의 DTD 구조에 맞게 변환하여 저장한다. 이 때도 역시 XSL 을 이용하여 XSLT 프로세스에 의해 변환작업이 수행된다. 마찬가지로 XConverter 시스템의 DTD 에 따른 XML 문서를 다른 XML 시스템으로 전송할 때 적용된다(<그림 9>의 (2) 참조).

<그림 10> HTML 용 XSL 예제

4. 결론 및 향후 연구과제

인터넷 기술이 발달하면서 각 사용자 별, 업무별로 다양한 메시지 표준들(ebXML, CML, WIDL 등)이 정의되어 사용되었다. 그러나 기존의 메시지 표준(EDI, UDF 등)을 위한 EDI 변환기(Translator)나 DB 를 위한 Mapper 로는 다양한 표준간 변환을 지원할 수 없었다. 그래서 새로운 메시지 표준이 정의될 때마다 새로운 변환기를 도입해야 하는 문제점이 제기되었으며, 사용자는 시스템 자원과 중복 투자 등 비용 부분에 대한 부담이 증가되었다. 이를 위해 본 논문에서는 어떠한 형태의 메시지가 전송되더라도 메시지 표준에 대한 매핑 정보만 등록함으로써, 사용자가 요청한 형태로 변환할 수 있으며, 기존 Legacy System(ERP, MIS 등)과도 연동 가능한 XConverter 시스템을 설계하였다. 그리고 제공할 수 있는 여러 메시지 형태 중 EDI, XML, DB, HTML 등 일부를 우선 선정하여 구현하였다. 향후 연구과제로는 기업간 전송되는 메시지 형태(예 : EDI, XML, UDF 등)에서 일반 사용자들이 사용하는 메시지 형태(예 : Excel, Word 등)까지 포괄적으로 지원할 수 있도록 확장성 있는 시스템으로, 또한 기존의 RDBMS 와 연동 가능한 DB 변환 모듈을 Component 화하여 사용자 Legacy System 내 모듈로서 동작하도록 구성하겠다. 마지막으로 인터넷을 사용함으로 인해 여러 보안 위협 요소에 노출되어 있는데, 특히 기업간 또는 개인간 전송되는 문서 중에는 기업의 중요 기밀이나 비용에 관한 내용이 존재할 수 있어 보안이 중요시 되고 있다. 그러므로, 메시지 암호화나 전자서명을 적용하여 안전하고 신뢰할 수 있는 서비스를 제공하는 기반 구조를 구축하도록 하겠다.

참 고 문 헌

[1] [성균관대, 1993] 한국통신 최종 연구 보고서, 성균관대, "EDI 시스템 시큐리티 선행기술 연구", 1993.

[2] [안경림, 1994] "OSI 환경을 위한 EDI 보안서비스요소의 설계 및 구현", 석사 논문, 1994.

[3] [안경림, 박상필, 안정희, 2001] "인터넷을 기반으로 하는 메시징 시스템(XML/EDI System) 설계 및 구현", 한국정보거래(CALS/EC)학회지 제 5 권 제 2 호, 2001. 3, pp.101-112

[4] [KEB, 1993] "EDI & Security", (사)한국 EDIFACT 표준원 1993. 5.

[5] [KTNET, 1995] "전자문서 표준안", (주)한국무역정보통신 1995. 5.

[6] [Dan Chang&Dan Harkey, 1998] Dan Chang, Dan Harkey : Client/Server Data access with Java and XML, Wiley & Sons Inc., Canada, 1998

[7] [Sean McGrath, 2000] Sean McGrath:XML Processing with Python, Prentice-Hall. Upper saddle River, NJ. 2000.

[8] [David Webber, 1998] David Webber: XML/EDI Perspectives, Japan.

[9] [Martin Bryan, 1998] Martine Bryan, Beno Marchal, Norbert H Mikula, Bruce Peat and David RR Webber : Guidelines for using XML for Electronic Data Interchange. W3C XML/EDI Group. 25. Jan. 1998.

[10] [XML-W3C Recommendation, 1998] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen W3C. 6 Oct 1998. : Extensible Markup Language (XML) 1.0 Specification,

[11] [XSL- W3C Recommendation, 2000] Sharon Adler,Anders Berglund, Jeff Caruso, Stephen Deach, Paul Grosso, Eduardo Gutentag, Alex Milowski, Scott Parnell, Jeremy Richman, Steve Zilles. W3C 21 Nov 2000. : Extensible Stylesheet Language (XSL) Version 1.0.

저자 소개

안경림

충북대학교 컴퓨터공학과 학사
성균관대학교 정보공학과 석사
현재 한국물류정보통신㈜ 연구소
관심분야 : 메시징 시스템(X400), 전자상거래, E-Maketplace, 무선통신, 보안 등

백해경

숭실대학교 전자계산학과 학사
현재 한국물류정보통신㈜ 연구소
관심분야 : 전자상거래, E-Maketplace, C-Commerce, 보안 등

임병찬

한양대학교 전자.전자통신.전파공학과군 학사
현재 한국물류정보통신㈜ 연구소
관심분야 : 전자상거래, E-Maketplace, C-Commerce, 보안

이영교

한양대학교 전자공학과 학사
한양대학교 전자공학과 석사
대우통신 종합연구소 선임연구원
현재 LG 전자 중앙연구소 선임연구원
관심분야 : 교환기, 통신, 전자상거래 등