

B2B 환경에서 XML을 활용한 EDI 시스템의 설계 및 구현

한순희, 정덕길
동의대학교 전산통계학과

Design and Implementation of the EDI System Using XML on B2B

Soon-Hee Han, Deok-Gil Jung
Dept. of Computer Science & Statistics, Dong-Eui University
E-mail : 99kb009@hanmail.net, dgjung@hyomin.dongueui.ac.kr,

요 약

세계는 개방화되어 가고 있으며, 이에 따라 정보의 가치가 더욱 중요하게 되고 있다. 국가와 기업이 경쟁에서 살아남기 위해서 정보의 효율적인 관리와 공유, 그리고 가치 있는 정보로 가공하는 것이 필요하다. 이러한 환경에서 등장한 것이 EDI이다. XML/EDI는 전통적인 EDI의 단점을 해결하며 표준의 한가지 접근방법으로 개방적이고 사용자간의 상호 대화가 가능하며 객체지향적이라는 특징을 가지고 있다. 본 논문에서는 XML을 EDI에 도입함으로써 대규모의 전자 상거래의 자동화를 가능하게 하며 고비용, 저효율적인 EDI 문제점을 해결하는 XML을 활용한 EDI 시스템의 설계 및 구현의 한 방안을 제시한다.

1. 서론

국내외를 막론하고 초고속정보통신망의 구축작업이 활발히 진행되고 있으며, 정보통신망을 이용하여 정보를 교환하고 업무의 효율화를 꾀하려는 노력이 활성화되고 있다. 이러한 노력은 기업 내부는 물론 외부와의 신속, 정확한 정보의 교환과 공유가 이루어질 때 가능하며, 현재 쓰이고 있는 대표적인 방안이 전자문서교환(EDI, Electric Data Interchange)시스템이다.[8]

EDI는 데이터 자체가 고정되어 있고 정형화되어 있어서 다른 시스템과의 호환 시에 많은 문제점을 가지며 구조적인 데이터의 검색 및 처리에 어려움이 있다. 또한 기업간의 데이터교환을 위해 VAN(Value-Added Network)을 이용하여 데이터를 전달했으나, VAN의 폐쇄성으로 인하여 시스템의 확장에 문제가 제기되었다. 이에 따라 본 논문에서는 VAN을 이용하지 않고 기존에 구축되어 있는 웹을 기반으로 구조적이고 확장성이 뛰어난 인터넷 표준언어인 XML(eXtensible Markup Language)을 이용하여 EDI문서를 처리하기 위한 시스템을 설계하고 구현한다.

2. XML/EDI 관련연구

2.1 EDI

EDI의 주요 구성요소는 다음과 같다.

(1) EDI의 표준 EDIFACT

행정, 무역 및 운송 업무에서 교환되는 전자문서(메시지)의 내용과 구조, 통신방법 등에 관한 일련의 규칙을 제공한다.

(2) 사용자 시스템

거래 당사자간에 데이터 통신망을 통해 전자문서를 주고받기 위해 사용자가 갖추어야 할 하드웨어, 소프트웨어, 통신장비를 말한다.

(3) 통신 네트워크(VAN)

거래 당사자들이 전자문서를 주고받는 통신방법, 통신시간, 통신속도들이 상이하므로 이를 통합관리하여 중계, 전송해주는 역할을 수행한다.

2.2 XML

XML은 1998년 2월 W3C(World Wide Web Consortium)에서 제정한 표준으로서, 웹에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이다. XML은 문서에서 사용되는 태그의 정의와

태그들 사이의 관계를 정의함으로써 문서의 구조를 정의하는 DTD(Document Type Definition)를 이용하여 사용자의 용도에 맞는 문서를 작성할 수 있다. 또한 XSL(XML StyleSheet Language)을 사용하여 문서의 표현논리를 별도로 분리시켜 저장할 수 있다.

XML의 주요 특징으로는 문서구조, 데이터 및 표현 부분을 따로 분리시켜 데이터를 가공하는데 유연성과 확장성을 제공한다는 것이다. 따라서 이기종간에 문서 교환이 요구되거나, 문서 정보 처리를 웹 브라우저로 전송할 때, 동일한 문서를 사용자의 요청에 따라 다른 형식으로 보여 줄 때 혹은 사용자의 요청에 따라 문서, 정보 수집을 특성화시켜야 할 때 사용할 수 있다.

2.3 XML/EDI

1996년 제6회 WWW 컨퍼런스에서 XML이 발표된 이후 WEB을 포함한 많은 분야에서 XML을 응용한 시스템을 개발하여 활용하려는 노력이 활발히 진행되고 있다. XML/EDI는 EDI 표준을 DTD로 표준화하여 이를 기본으로 문서를 작성하게 되므로 기업들 간에 표준화된 문서 교환을 가능하게 해준다.

전통적인 EDI에서는 데이터를 분리하고 인식하기 위해 토큰과 같은 세그먼트 인식자를 가지고 있다. XML은 그러한 세그먼트를 웹 토큰으로 치환함으로써, XML이 EDI문서를 표현할 수 있으며, EDI 문서를 HTTP나 SMTP와 같은 웹을 통한 전송매체를 이용하여 전달할 수 있다. XML과 EDI가 통합됨으로써, 사용자들은 기존의 서로 다른 EDI 소프트웨어의 사용에 따른 불편함을 해소할 수 있으며, 통합된 환경에서 작업을 수행할 수 있다.

이와 같이 XML을 활용하여 전통적인 EDI의 단점을 극복하고, EDI를 웹 기반으로 확장시키는 전자상거래 프레임 워크를 제시하고 있다.

2.4 XML/EDI 시스템 구현의 요구사항

일반적인 XML 문서와 EDI 문서의 차이는 DTD와 XSL에서 나타나며, 일반적인 XML 문서는 사용자가 태그를 확장하는 것이 가능하지만, EDI에서는 국제표준으로 정의된 값이 반드시 이용되어야 하므로 사용자가 EDI망과 연계를 원한다면 그 규칙을 지켜주어야 하기 때문에 DTD와 XSL 설계에 신중을 기해야 한다.

XML/EDI 가이드라인에 관한 기준과 저장소(repository)에 관한 표준[4] 등이 제안되고 있으나 아직까지는 EDI 표준화 기관인 UN/EDIFACT에서 공식적인 표준으로 채택하지 않고 있다. 본 논문에서 구축하고 있는 XML/EDI 시스템 구현의 요구사항은 다

음과 같다.

- ① 기존 시스템 활용을 보장해야 한다.
- ② XML을 교환문서의 표준으로 활용하는 EDI 시스템을 구성해야 한다.
- ③ XML/EDI 표준이 확정되고 상용화될 때를 대비한 확장성과 유연성을 가지고 있어야 한다.
- ④ 어떠한 데이터 형식으로도 접근 가능해야 한다.

2.5 XML/EDI 시스템의 파급효과 및 활용방안

본 논문에서 설계하고 구현하는 XML/EDI 시스템을 통하여 전자상거래 및 물류관리 시스템의 엔진으로 이용할 수 있으며, 다음과 같은 파급효과 및 활용방안이 있다.

- ① 별도의 전용 EDI 소프트웨어 없이 인터넷 표준 브라우저만으로 이용 가능하다.
- ② EDI가 가진 단점인 문서 형식 변경의 어려움을 극복할 수 있다.
- ③ VAN 사업자가 전담 매개체에서 부가 서비스 및 통합환경 제공자로 변화된다.
- ④ 기업 내에 설치된 IT 시스템들을 서로 연동하는 기반 시스템으로 이용 가능하다.
- ⑤ CALS 구축 시에 통합문서 교환시스템으로 활용할 수 있다.
- ⑥ HTML 문서 및 기타 관련문서의 관리에 활용함으로써 EDI 문서와 HTML 문서간의 변환기능을 제공하여 인트라넷과 전자상거래의 결합을 유도한다.
- ⑦ 그룹웨어 개발 시에 표준문서의 관리 및 검색 도구를 제공할 수다
- ⑧ 기존에 사용 중인 전자문서 관리시스템과 연동하여 문서의 교환 및 관리시스템으로 활용할 수 있다.

3. XML/EDI 시스템

3.1 XML/EDI 시스템의 개요

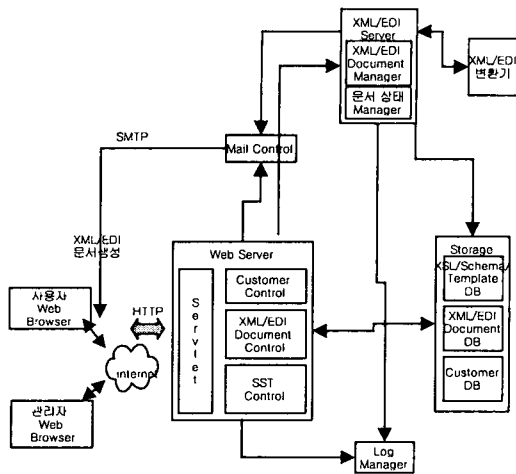


그림1. Internet 기반 XML/EDI 시스템의 전체구조

그림1에서 볼 수 있는바와 같이 Internet 기반 통합 XML/EDI 시스템은 문서를 교환할 수 있는 웹서버와 전역저장소(Global Repository)의 역할을 하는 웹서버 그리고 데이터베이스 서버가 필요하다.

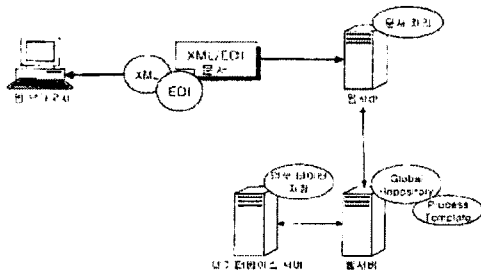


그림2. 웹을 기반으로 한 통합 XML/EDI 트랜잭션모델

그림2에 표시되어 있는 거래 상대방 간에 발생하는 XML/EDI 시스템 구조에서는, 거래 양측에서 참조하는 DTD, XSL, Template 등은 거래 당사자 간에 전역저장소에 존재하며 거래 상대방들이 공유하며 사용한다.[2] 거래 상대방간 사용문서에 대한 DTD를 가지고 있으며 업무규칙과 문서의 형태를 정의하는 XSL, 전송 데이터의 처리 내용을 담고 있는 Template을 참조하여 전송한다. 수신 측에서는 전용 브라우저나, HTML등을 활용하여 전송된 문서를 화면에 출력한다.[4, 5, 7]

(1) XML 파서(parser)

XML 파서는 XML 문서 처리의 기반이 되는 핵심

기술로서 XML 문서를 해석하고 필요한 정보를 추출하며 오류를 점검하는 기능을 제공한다.

(2) XML 문서의 저장

구조화 문서의 저장 관리는 파일을 이용하는 정보 검색 시스템과 데이터베이스를 이용하는 시스템으로 분류된다. 일반적으로 XML 문서의 구조 정보를 추출하여 관계 데이터베이스의 테이블이나 객체지향 데이터베이스의 객체를 이용하여 저장하게 된다. 다양한 문서들은 사용 환경이나 목적에 맞게 데이터베이스 시스템에 저장되어 관리되는데 일반적으로 데이터베이스 시스템은 정형화되고 구조화된 데이터에 유리하다.[6]

그림3은 XML 파서에 의해 오류를 점검한 후 관계 데이터베이스와 객체지향 데이터베이스에 저장되는 과정을 나타낸다

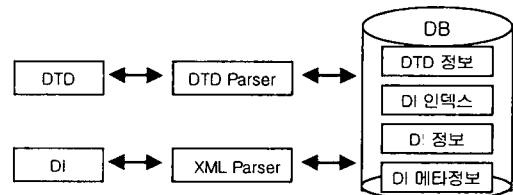


그림3. XML 파서와 문서 데이터베이스와의 관계

본 논문에서 구현한 시스템은 EDI 문서의 특성을 고려하여 기존의 정보시스템에서 사용하는 파일 저장 방식과 관계형 데이터베이스 저장방식을 혼용하는 하이브리드 저장시스템을 구현하였다.

하이브리드 시스템의 장점은 다음과 같다.

- ① 엘리먼트 단위로 저장을 함으로써 광범위한 문서에 대해 정확한 질의를 통해 원하는 문서를 쉽게 찾을 수 있다.
- ② 파일로 저장된 특정 문서에 대해서도 질의가 가능하므로 검색의 정확성과 효율을 높일 수 있다
- ③ 사용자에게 문서를 보여주기 위한 XSL의 적용이 필수적인데 만일, 데이터베이스에 저장된 엘리먼트를 재조립하여 XML문서를 생성해야하는 경우 과도한 Join 연산으로 데이터베이스의 과부하를 초래할 수 있으므로 파일에 저장된 XML문서를 이용한다.

3.2 XML 문서 저장 모듈

XML 문서 저장 모듈의 전체적인 구조는 그림4와 같이 사용자 응용프로그램 부분인 Application Layer

와 실질적인 문서저장 부분을 담당하고 있는 Store Layer 그리고 문서 템플릿, DTD, XSL, 문서메타-색인정보, XML 문서와 그 문서를 엘리먼트 단위로 저장하는 Fragment가 저장되는 Repository Layer의 3계층으로 구성되어 있다. 사용자는 XML 문법을 숙지하지 않더라도 웹의 입력 폼으로부터 문서생성에 필요한 데이터를 입력하여 문서를 생성하며, 이 때 문서 생성 모듈은 템플릿을 참조하여 새로운 문서 인스턴스를 생성하게 된다.

문서 저장 모듈은 생성된 문서를 파싱하여 데이터 메모리에 생성하고 문서의 구조적 정보를 추출하고 색인하여 데이터베이스에 문서를 엘리먼트 단위로 저장하게 된다.

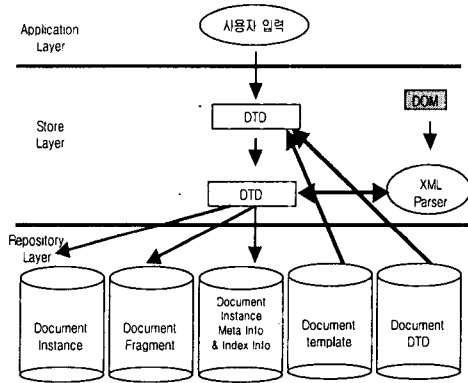


그림4. XML 문서 저장 모듈

4. XML 기반의 EDI 구현 프로세스

본 논문에서는 인터넷상에서 EDI를 구현함에 있어 XML을 사용하는 방법을 제안한다.

XML은 웹을 통한 검색이 가능하고 기존 EDI 시스템의 폐쇄성을 극복할 수 있으며, 문서의 논리적 확장성을 제공하므로 기존 EDI 표준 항목에 유연성을 부여할 수 있어 기존 EDI의 약점을 보완할 수 있다.

그리고 미리 정해진 EDI 표준에 데이터를 맞추기 위한 분석에 들어가는 비용 및 시간을 절감할 수 있게 된다. 또한, XML은 매우 구조적인 언어이다. EDI는 문서의 동일 형식에 내용만 달라지는 반복업무의 특성이 강하므로 XML로 표현하면 XML 문서를 파싱하여 실제 문서 내용만 데이터베이스에 저장함으로써 문서 전체를 저장하는 비효율을 극복할 수 있고 자료의 구조 검색이 가능하게 된다.

4.1 개발도구

(1) HTML : 인스턴스 생성 윈도우와 데이터 입력

화면인 사용자 인터페이스에 사용한다.

(2) Java1.2, Servlet, JSP : 인스턴스 생성 윈도우 유도 화면으로부터 인스턴스 생성 윈도우를 활성화 시키는 과정과 실제 인스턴스를 생성시키는 과정에 사용한다.

(3) JDBC : 각 인스턴스와 각각의 트랜잭션을 데이터베이스에 저장시키는 과정에 사용한다.[1]

4.2 구현 프로세스

실제 XML 문서로 EDI를 구현하는 프로세스는 그림5에 표시된 바와 같이 수행한다.

모든 프로세스는 웹 상에서 이루어지는데, 먼저 EDI표준항목을 분석하여 XML DTD를 작성한다. 작성된 각각의 DTD로부터 Java Servlet을 사용하여 인스턴스 생성 윈도우를 활성화 시키고, 활성화된 윈도우에 항목을 기입하면 실제 인스턴스가 생성될 수 있게 한다.

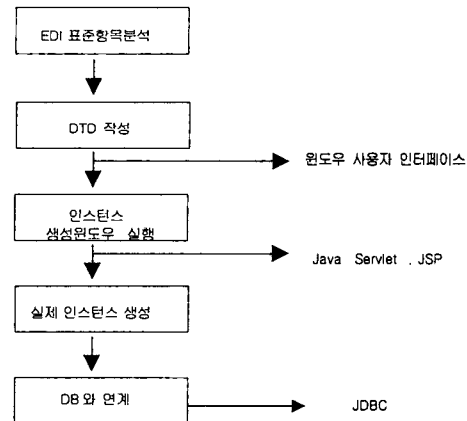


그림5. 구현 프로세스

실제 XML 인스턴스를 생성시켜 DB에 저장시키는 절차는 다음과 같다.

- ① 작성하고자 하는 EDI 문서의 DTD를 선택한다.
- ② 인스턴스 생성 윈도우를 활성화한다.
- ③ 인스턴스 생성 윈도우에 데이터를 기입한다.
- ④ 인스턴스를 생성한다.
- ⑤ JDBC를 이용하여 DB에 저장한다.

위의 과정①은 EDI 표준 항목에 대한 DTD가 만들어져 있다는 가정 하에 이루어진다.

5. XML을 활용한 EDI 문서처리 시스템 구현

앞에서 기술된 설계에 기반하여 구현된 XML/EDI 시스템의 전체 구조는 문서 생성 모듈, 문서 저장 모

들로 구성되어 있다.

5.1 XML/EDI 시스템의 개발 환경

XML/EDI 시스템의 개발환경은 다음과 같다.

- 운영체제 : Windows 2000
- 웹 서버 : Resin 2.0
- 데이터베이스 : MS-Access
- 사용언어 : Java 1.2, Servlet, JSP

5.2 문서생성 모듈 : XML 문서 인스턴스의 생성

문서생성모듈에서는 클라이언트로부터 전송된 Multipart/Form-Data 형식의 문자 스트림이 서버에 Post된다. 전송된 엘리먼트 콘텐츠에 해당하는 값을 추출하고, 동시에 템플릿 문서를 파싱하여 동일한 이름을 갖는 엘리먼트 콘텐츠를 치환함으로써 XML 인스턴스 문서를 생성한다.

5.3 문서저장모듈 : XML 문서 인스턴스 엘리먼트의 저장

문서저장모듈에서는 생성된 XML 인스턴스 문서를 파싱하여 엘리먼트와 그 경로에 대한 정보를 DB에 저장한다.

6. 시스템의 실행

6.1 주문서 EDI 항목의 DTD 작성

XML 문서 인스턴스를 생성시키기 위한 사전 작업으로 주문서에 관한 XML DTD를 표1과 같이 작성하였다.

표1. 주문서의 XML DTD

```

<!-- ***** OFFERSHEET.dtd ***** -->
<!ELEMENT OFFERSHEETTOP (OFFERSHEET)+>
<!ELEMENT OFFERSHEET (OURREF,ITEMS,SUBNOTES,
YOURREF)>
<!ELEMENT OURREF (OFFERNAME)>
<!ELEMENT OFFERNAME (#PCDATA)>
<!ELEMENT ITEMS (ITEMNO,COMMODITY,UNITPRICE
,AMOUNT)>
<!ELEMENT ITEMNO (#PCDATA)>
<!ELEMENT COMMODITY (#PCDATA)>
<!ELEMENT UNITPRICE (#PCDATA)>
<!ELEMENT AMOUNT (#PCDATA)>
<!ELEMENT SUBNOTES (ORIGIN,SHIPMENT,SHIPPINGPORT,
DESTINATION,REMARKS)>
<!ELEMENT ORIGIN (#PCDATA)>
<!ELEMENT SHIPMENT (#PCDATA)>
<!ELEMENT SHIPPINGPORT (#PCDATA)>
<!ELEMENT DESTINATION (#PCDATA)>
<!ELEMENT REMARKS (#PCDATA)>
<!ELEMENT YOURREF (RECIERVERNAME)+>
<!ELEMENT RECIERVERNAME (#PCDATA)>
    
```

6.2 인스턴스 생성 윈도우 활성화

문서 작성 화면은 HTML 폼 페이지로 구성되어 있으며, 사용자가 해당 텍스트 필드에 데이터를 모두 입력하고 Multipart/Form-Data 형식으로 전송하면 서버에서 데이터를, 파싱하여 XML DI를 생성하게 된다. 생성된 DI는 XML 파싱을 통해 문서의 구조적 정보가 DB에 저장된다.

그림에서는 XML 문서를 생성하기 위한 입력하는 화면을 보여주고 있으며, Post하여 서버 측에 저장하는 화면을 보여주고 있다.

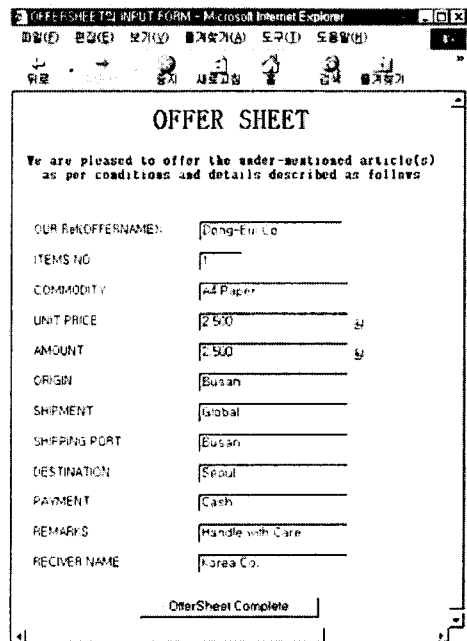


그림6. 주문서 작성화면(인스턴스 생성 윈도우)

적합한 DTD로 XML 문서를 검증하기 위해 DTD는 전역저장소에 저장된다.

그림7은 Post 된 데이터를 JSP에서 수신하여 XML 문서의 인스턴스화 한 화면이다. 생성된 XML 문서가 적합한 경우에, 인터넷 브라우저가 DTD를 참조하여 확인할 수 있다. 이때, XML 문법과 DTD에 부합되지 않으면 웹브라우저는 에러 메시지를 출력하게 된다.

그림7에 표시된 바와 같이 인스턴스화 된 XML은 JSP을 통하여 데이터베이스에 저장된다.

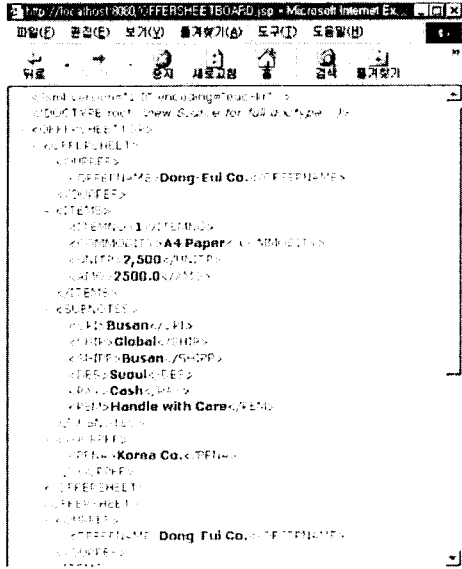


그림7. XML 인스턴스 생성 화면

6.3 데이터베이스와의 연계

그림8에서는 생성된 각각의 인스턴스와 트랜잭션들이 테이블 형태로 DB에 저장되는 화면을 보여주고 있다. 이 과정에서는 JDBC를 이용하여 MS-Access DB에 저장된다. 이와 같이 테이블 형태로 인스턴스를 저장하여 정보의 구조적 검색이 가능하게 된다.

ID	OPNA	ITNO	COM	UNITP	AMO	ORI	SHIP	S
1	Dong-Eul Co. I	A4 Paper	2,500	2500	Busan	Goba	EJS	
2	Dong-Eul Co. I	A4 Paper	2,500	2500	Busan	Goba	EJS	

그림 8 XML 인스턴스의 DB 저장화면

7. 결론

1990년대 후반기 들어 일기 시작한 인터넷의 열풍과 Client/Server 정보기술의 발달, CALS와 같은 표준규격의 확산과 더불어 EDI 분야에서도 전통적인 EDI방식에서 탈피하여 인터넷을 이용한 차세대 EDI의 실현에 대한 연구개발이 선진 각국을 주축으로 진행되고 있다. 이와 같은 추세는 물론 전통적인 EDI의 문제를 극복하기 위한 목적에 있다.

XML/EDI시스템에서는 데이터 전달에 초점을 둔 전통적인 EDI의 범위를 확대하여 전자상거래에 필요한 프레임워크를 제공한다는 측면에서 전통적인 EDI와는 그 개념이 상이하다. 따라서, 기존의 EDI 서비스

를 인터넷 플랫폼으로 이동시키고 다시 웹 플랫폼으로 이동시킴으로써 EDI 서비스 사용자들은 기존의 서로 다른 EDI 전용 소프트웨어의 사용에 따른 불편함이 없어지고 통합된 환경에서 EDI 서비스를 사용할 수 있다.

본 논문에서 제시한 XML/EDI시스템 구현 후 다음과 같은 방향으로 계속해서 연구를 진행할 예정이다.

- ① 문서의 수신 및 검색 모듈에 관한 연구
- ② 효과적인 전역저장소의 이용 방법 연구
- ③ 웹 상에서 문서관리에 관한 보안 연구
- ④ EDIFACT 문서의 적용 및 변환기 적용
- ⑤ 차세대 XML 기반의 통신 프레임워크인 SOAP (Simple Object Access Protocol)의 지원으로 타 IT 시스템 및 XML/EDI Server간의 연동[3]

[참고문헌]

- [1] Hamilton, G. and Cattell, R., *JDBC: A Java SQL API*, JavaSoft INC., 1996.
- [2] Natanya Pitts, Lee S, J, *XML 21일 완성*, 삼각형 프레스, 1999.
- [3] The XML Cover Pages, "Simple Object Access Protocol (SOAP)", <http://www.oasis-open.org>.
- [4] XML/EDI Group, "XML/EDI Transaction Models," <http://www.geocities.com/WallStreet/Floor/5815>
- [5] 김우현, 전자서명 알고리즘을 적용한 암호화된 XML/EDI 구현, 대구대학교 대학원, 2000.
- [6] 김영일, 객체지향 데이터베이스를 이용한 XML 문서 저장시스템설계 및 구현, 석사학위논문, 충남대학교, 1999.
- [7] 박성득, *SGML, XML, EDI 통합 및 연계방안*. 한국전산원, 1999.
- [8] 정희경, *XML가이드*, 도서출판 그린, 1999.