

웹 기반 XML/EDI 시스템 설계 및 구현*

구태희,** 변광준,*** 황인준***

An Implementation of Web-based XML/EDI System

Tae-Hoe Koo, Kwang-June Byeon, Een-Jun Hwang

Abstract

The EDI system supports the working environment which can exchange the standardized document through the each agreed communication method. Traditional EDI system is complicated to be implemented and needed high cost to using the system because of the specific VAN(Value Added Network). This paper proposes the XML/EDI system using the web-based distributed object and proposed the component 'AnySchema', as it makes be able to use legacy Database for EDI user. This XML/EDI system exchanges the XML documents through the Internet, so proposed low cost to support the system. Also, by using the XML document, it doesn't need EDI specific software and easy to exchange data between different systems. This system is implemented by using Java, as platform-independent.

* 본 논문은 1999년 3월부터 2000년 2월까지 ㈜대우정보시스템과 진행한 산학프로젝트를 구현모델로 한 것임.

** (주)COTECH 사

*** 아주대학교 정보통신전문대학원 정보통신공학과, 정보 및 컴퓨터공학부

1. 서론

기업, 공공기관 등은 50년대부터 조직 내부의 업무 처리에 소요되는 시간, 인력, 비용을 절감하기 위하여 컴퓨터를 도입하기 시작하였다. 이러한 조직들은 타 기관과 정보 및 문서를 교환하기 위하여 일반적으로 우편, 전화, (40년대부터) 텔렉스, (80년대부터) 팩스 등을 이용하였다.

조직 내부의 업무처리가 상당 부분 전산화되면서 문서 발신처는 컴퓨터에 입력하여 처리한 자료를 출력하여 우편으로 발송하고, 수신처는 전달된 자료를 내부에서 전산 처리하기 위하여 컴퓨터에 다시 입력하는 것이 일반적이었다. 이러한 업무처리방식은 자료의 재입력 과정에서 다수의 오류가 발생하며, 따라서 많은 시간, 인력, 비용을 수반한다.

이와 같이 출력된 자료의 재입력을 생략하고 컴퓨터간에 자료를 교환하여 그대로 활용하기 위하여 종이 카드, 종이 테이프 등에 출력하여 전달하거나, 마그네틱 테이프, 디스켓 등에 저장하여 교환하기 시작하였다. 그러나 자료의 전자적인 통신(data communication) 수단이 발달하고 통신비용이 저렴해지면서 전자적인 자료의 교환에 전자통신매체를 활용하기 시작하였다.[2]

EDI(Electronic Data Interchange)를 이용하여 얻을 수 있는 효과로는 처리시간 단축, 업무부대비용 감소, 업무오류 방지, 물류비 절감, 이미지개선, 인력절감을 들 수 있다. 그러나, 이러한 EDI의 효과에 비하여 현재 EDI를 활용하는 비율은 낮은 것으로 나타나고있다. 1998년 상시종업원 10인 이

상 수도권지역 800개 사업체를 대상으로 (재)한국전자거래표준원에서 조사한 바에 의하면 현재 EDI를 사용하는 기업은 31%에 그치고 있으며, 17.6%의 기업만이 향후 도입을 고려하고 있는 것으로 나타나고 있다. 이 보고서에 따르면, 그 중요성에 비하여 이와 같이 도입을 저해하는 대표적 요인으로는 사용자 내부의 응용 Software와의 연계어려움, EDI Software의 폐쇄성, 법령 및 제도의 미비 등인 것으로 조사되었다. 소규모 기업들의 경우 기업의 규모에 비하여 과다한 도입 및 운영비용 역시 EDI확산에 걸림돌이 되고 있는 실정이다.

1990년대 후반 들어 불기 시작한 Internet의 열풍과 Client/Server정보기술의 발달, CALS(Commerce at Light Speed)와 같은 표준규격의 확산과 더불어 EDI분야에도 전통적인 EDI방식에서 탈피하여 Internet을 이용한 차세대 EDI의 실현에 대한 연구개발이 선진각국을 주축으로 진행되고 있다. 이의 목적은 물론 위에 언급한 전통적인 EDI의 문제를 극복하기 위해서이다. 즉, 중소기업에 까지 확산 가능한 경제적이며, 개방된 구조의 EDI를 실현하는 것에 그 목적이 있다. 1998년 3월에 ISO 8879인 SGML의 단순화된 버전인 XML(eXtensible Markup Language)이 국제적인 표준으로 채택된 후에는 이를 기반으로 XML/EDI의 연구개발이 가속화 되고 있다.[3]

리서치그룹인 GartnerGroup과 Dataquest에서는 EDI 시장의 장기 전망에서 기존의 전통적인 EDI 소프트웨어와 서비스 시장의 성장은 둔화되는 반면, 인터넷/웹 기반의 EDI 시장의 성장은 가속화될 것으로 내다

보았다. 아래의 <표 1>, <표 2>에는 지난 3년간의 EDI소프트웨어와 서비스 시장의 현황에 대해 나타내고 있다.

<표 1> 최근 3년간의 EDI S/W 시장현황
(단위: 백만달러)

| 구분\연도 | 1996 | 1997 | 1998 |
|-----------------------|-------|-------|-------|
| 기존의 EDI 소프트웨어 매출액 | 352.8 | 380.6 | 382 |
| 기존 EDI S/W 연성장률(%) | | 7.9 | 0.4 |
| 인터넷/웹 EDI S/W 매출액 | 0 | 11.4 | 45.2 |
| 인터넷/웹 EDI S/W 연성장률(%) | | | 297.6 |

<자료> Dataquest, 1999.5

<표 2> 최근 3년간의 EDI VAN 서비스 시장현황
(단위: 백만달러)

| 구분\연도 | 1996 | 1997 | 1998 |
|--------------------------|-------|-------|--------|
| 기존의 EDI VAN서비스 매출액 | 801.7 | 933.5 | 1003.4 |
| 기존 EDI VAN서비스 연성장률(%) | | 16.4 | 7.5 |
| 인터넷/웹 EDI VAN서비스 매출액 | 0 | 11.5 | 52.4 |
| 인터넷/웹 EDI VAN서비스 연성장률(%) | | | 355.0 |

<자료> Dataquest, 1999.5

이러한 환경에서 VAN/EDI 사용자가 XML/EDI시스템을 도입하더라도 기존의 데이터베이스의 데이터를 이용하는 내부 응용 프로그램을 위해서 기존의 데이터베이스에 수신된 XML 문서의 데이터를 갱신(update)시켜 주어야 한다. 그럼으로써, 시스템을 도입한 기업에서 XML/EDI시스템을 도입함으로써 기존의 데이터베이스와 관련 응용프로그램의 손실을 피할 수 있게 해 주는 것이다.

2. 관련연구

2.1 EDI 개념

정보화 사회가 진전됨에 따라 급속도로 늘어나는 거래 정보 교환을 신속하고 정확히 처리하기 위해 출현한 것이 EDI이다.

우리나라 무역자동화법에 의하면 EDI란 “업무를 컴퓨터 등 정보 처리능력을 가진 장치간에 전기 통신 설비를 이용하여 전자 문서로 전송 처리 또는 보관하는 방식”이라고 정의하고 있다.(무역자동화법 2 조 6 호) 한편, 컴퓨터 용어 사전에 의하면 EDI란 “전자 데이터 교환”이라고 하며 때로는 “전자 자동 거래”라고도 한다. 전자 데이터 교환이란 기업간의 거래에 관한 데이터(각종 서류 양식)를 정형화하여 컴퓨터 통신망(Computer Network)을 통해 거래 당사자의 컴퓨터 사이에서 직접 정보를 주고 받는 것을 말한다.

정형화된 형식에 맞추어 구성 조합한 데이터를 사용자간 문서방식의 불일치를 줄이거나 없애기 위하여 합의된 문서표준으로

바꾸어, 한 컴퓨터 시스템에서 다른 시스템으로 전자적 통신 수단을 활용하여 문서를 전달함으로써 자료 입력 비용의 감소, 서류 제작성 업무의 감소, 기록의 정확성을 도모할 수 있게 되는 것이다.

EDI의 개념을 좀더 명확히 하면 다음과 같다.

(1) EDI는 조직 대 조직(기업과 기업, 혹은 기관과 기관, 기업)간의 정보 전달을 의미한다.

(2) 통상적인 의미에서의 EDI는 사람의 개입이 없이 응용 시스템간에 데이터 교환이 일어나고 교환된 데이터는 상대측 응용 시스템에서 이용되어야 한다.

(3) EDI에서 교환되는 데이터는 구매 요청서나 송장과 같이 구조화된 데이터이지만, 문서나 서류상의 레이아웃 정보는 포함하지 않는다.

(4) EDI는 전자 우편과 달리 전자적 계약 및 법적 구속력이 성립된다. 이를 위해 보안 서비스가 필수적으로 적용되어야 한다.

2.2 EDI 표준

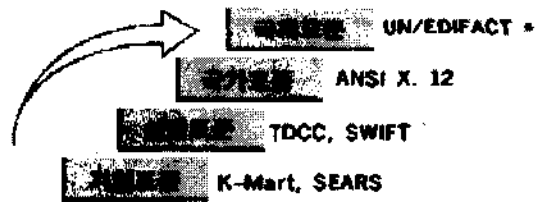
1960년 처음 도입되었던 초창기의 EDI는 개별 기업 내지는 그룹 차원에서 사설 표준을 기반으로 컴퓨터를 연결시켜 정보를 교환하였다.

그러나 동일 산업내의 타기업과도 정보 교환을 해야 할 필요성이 나타나자 사설 표준의 한계를 인식하고 산업 표준을 제정하게 되었다. 예를 들면 유럽 자동차 산업을 위한 ODETTE(Organization for Data Exchange by Teletransmission in Europe), 영국과 아일랜드

드의 소매 산업을 위한 TradaNet이 있다.

그 후 경제가 지속적으로 발전하면서 전국의 모든 산업간 거래가 활발해지자 국가 표준을 제정하게 된다. 북미의 ANSI ASCX12(American Standards Institute Accredited Standards Committee)와 유럽의 GTDI(Guidelines for Trade Data Interchange)가 그것이다.

1980년대 들어 기업의 국제화, 개방화가 급속히 진행되면서 국제 표준의 필요성이 증대되자 1987년 UN의 후원 아래 미국과 유럽의 표준화 기구들이 모여 EDI 국제 표준인 UN/EDIFACT(The United Nation/Electronic Data Interchange for Administration, Commerce and Trade)를 제정하게 된다.



<그림 1> EDI 표준의 발전

2.3 XML 개요

XML은 1996년 W3C의 후원으로 형성된 XML Working Group에 의해 개발되었다. 1998년 2월 8일에 W3C의 Recommendation에 따라 XML1.0이 발표되었으며, 이 표준 규격은 문자에 관련된 표준인 ISO 10646 Unicode, 언어 인식 엘리먼트 표준인 Internet RFC 1766, 국가 언어명 코드를 위한 표준

ISO 639, 국가 명 코드를 위한 ISO 3166 과 관련하여 XML을 처리할 컴퓨터 프로그램을 설계하는 데 필요한 정보를 제공한다.

<표 3> XML의 역사

| |
|---|
| 1996.11. 보스턴, SGML 96 conference, XML을 위한 최초의 언어 규약 발표. |
| 1997. 3. GCA주최 샌디에고, 제 1 회 XML Conference 개최 |
| 1997. 4. XML 링크관련 규약 초안 제기 |
| 1997. 6. XML, C의 표준화로 공식화 됨 |
| 1997. 8. XML 문법관련 규약 초안 갱신 |
| 1997. 8. XSL 제안안, W3C에 제출 |
| 1997. 8. 몬트리올, XML Developers Day 개최 |
| 1997. 9. WIDL 제안안, W3C에 제출 |
| 1997. 9. 시드니, SGML/XML Asia Pacific 97 개최 |
| 1998. 2. XML 1.0 Recommendation 발표 |

XML은 SGML의 부분집합으로, SGML과 XML간의 변환이 용이하고 XML 문서를 SGML 어플리케이션에서 사용할 수도 있다. 또 XML의 eXtensible이란 말에서 알 수 있듯 태그의 확장이 SGML처럼 자유롭고 따라서 복잡하고 구조화된 문서에도 적합하다. 따라서 XML은 인터넷 상에서 의학, 경영, 법률, 논문 등 복잡하고 구조화된 문서 자료의 저장 및 관리, 검색을 용이하게 할 수 있고, 나아가 전자상거래, 디지털 도서관 등에서 사용될 것으로 전망되고 있다.[1, 4, 9, 10, 11, 12]

<표 4> XML 예제

```
<?xml version="1.0" encoding="euc-kr"?>
<?xml:stylesheet type="text/xsl" href="order2.xsl"?>
<!DOCTYPE OrderSheet SYSTEM "order.dtd">
<OrderSheet>
<HeaderInfo>
<DocName>*ORDERS</DocName>
<DocNum>128568</DocNum>
<DocCode>220</DocCode>
<ONum>0</ONum>
</HeaderInfo>
<PInfo>
<PCode>C</PCode>
<PName>CDMA</PName>
<OTotPrice>54000</OTotPrice>
<OTotQty>230</OTotQty>
</PInfo>
<ExInfo>
<CurCode>원</CurCode>
</ExInfo>
<SetInfo>
<SetCond>KEB4277</SetCond>
<SetDura>After30Days</SetDura>
<BoSet>A16</BoSet>
<AoSet>098-21-0450-999</AoSet>
</SetInfo>
.....
</OrderSheet>
```

2.4 DTD

DTD(Document Type Definition)는 문서에 사용될 데이터 엘리먼트를 위한 일련의 구문 규칙이다. DTD 문서 안에는 원하는 이름의 엘리먼트를 쓸 수 있고, 엘리먼트들이 XML 문서 안에서 어떤 순서로 나타나야 하며, 어떤 속성(attributes)을 갖는지 등을 말해준다.

XML은 언어 그 자체가 아니고 오히려 언어들을 정의하는 체계(system)이기 때문

에, XML은 HTML이 가지고 있는 것과 같은 보편적인 DTD를 갖지 않는다. 하지만, 데이터 교환을 위해 XML을 사용하고자 하는 기업이나 단체는 그 자신의 독자적인 DTD들을 정의할 수 있다.

<표 5> DTD 예제

```
<?xml version='1.0' encoding="UTF-8"?>
<!-- This Generator Test DTD-->

<!ELEMENT OrderSheet (HeaderInfo, PInfo,
ExInfo, SetInfo, PurInfo, SupInfo,
ItemInfo+)>

<!ELEMENT HeaderInfo (DocName,
DocNum, DocCode, ONum)>
<!ELEMENT DocName (#PCDATA) >
<!ELEMENT DocNum (#PCDATA) >
<!ELEMENT DocCode (#PCDATA) >
<!ELEMENT ONum (#PCDATA) >

<!ELEMENT PInfo (PCode, PName,
OTotPrice, OTotQty)>
<!ELEMENT PCode (#PCDATA) >
<!ELEMENT PName (#PCDATA) >
<!ELEMENT OTotPrice (#PCDATA) >
<!ELEMENT OTotQty (#PCDATA) >

<!ELEMENT ExInfo (CurCode)>
<!ELEMENT CurCode (#PCDATA) >

<!ELEMENT SetInfo (SetCond, SetDura,
BoSet, AoSet)>
<!ELEMENT SetCond (#PCDATA) >
<!ELEMENT SetDura (#PCDATA) >
<!ELEMENT BoSet (#PCDATA) >
<!ELEMENT AoSet (#PCDATA) >

.....
```

DTD는 어떤 문서의 형식에 대해서 공식적 정의를 담고 있는 하나의 파일(또는 함께 사용되는 복수의 파일들)이다. DTD에서 요소들의 이름, 요소들이 언제 나타나는지, 어떻게 함께 사용되는지 등을 정한다.

DTD는 텍스트를 포함하는 아이템과 아이템들을 포함하는 리스트를 정의한다. DTD는 언급한 대로 프로세서가 자동적으로 문서를 파싱하고, 모든 요소들이 언제 나오고 어떻게 서로 연관되어 있는지 구분할 수 있도록 하는 공식적인 언어이다. 그럼으로써 스타일시트, 네비게이터, 브라우저, 검색 엔진, 데이터베이스, 인쇄루틴 및 다른 응용 프로그램들을 사용할 수 있도록 한다[4, 9].

2.5 XML/EDI 개요

모든 거래 당사자에게 확산 가능한 경제적이며, 개방된 구조의 EDI를 실현코자 도입되기 시작한 것이 XML 기반의 EDI(이하 XML/EDI)이다. 특히 XML의 손쉬운 웹 확장을 통해 웹기반의 사용자 인터페이스가 동반된다.

1997년 인터넷 기반 상거래 확산 자원을 요청한 대통령의 요구에 따라 미국에서는 산업체와 학계가 참여하는 XML/EDI Group이 형성되어 주제별 토론과 정보교환, 표준화 방안 제안 등이 활발히 이루어지고 있으며, 유럽에서도 XML EDI 문서의 실용적인 전달에 관한 European XML/EDI 파일럿 프로젝트[CEN/ISSS XML/EDI Project Group 1999], EDIFACT의 PRICAT 메시지를 XML로 변환하는 XCAT 프로젝트 [Danish EDI Council 1998] 등이 진행 중이다.

2.6 XML/EDI 시스템

XML/EDI 시스템은 데이터 전달에 초점을 둔 전통적인 EDI의 범위를 확대하여 전자상거래에 필요한 Framework을 제공한다. 즉 상인과 고객간에 발생하는 "상품 전달을 통한 가치의 교환"을 위해 영업, 수배송, 물류, 수금, 세금처리, 생산 연계 등의 광범위한 활동을 통합적으로 지원하는 데이터의 교환 방식 및 시스템 Framework을 의미하게 된다. 이는 EDI를 XML의 Tag 사용해서 확장하는 것보다 훨씬 넓은 의미를 포함한다.

이러한 XML/EDI는 실행을 위해서 "데이터 교환 모델"을 위하여 XML을 사용하고, "모습을 표현하기 위하여" XSL(XML Style Language)을 이용하며, DTD를 사용하여 전통적인 EDI와 쉬운 통합방안을 가지며, 인터넷 프로토콜인 IP routing, HTTP, FTP 그리고 SMTP를 지원하고, 문서중심의 조회와 처리절차를 가능케 하며, CGI, Java와 ActiveX를 사용하여 프로그램을 개발하게 되는 것이다.

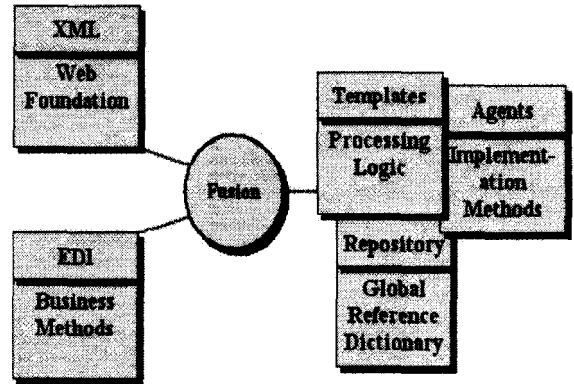
XML/EDI는 문서의 정보를 이전의 어떤 구조를 사용하는 것보다 더 정확하고 세밀하게 인코딩(Encoding)할 수 있다는 것이 가장 근본적인 특징이다. XML 태그와 DTD를 통해서 XML/EDI 데이터는 스스로를 설명하는 정보를 포함하므로 문서가 트랜잭션(Transaction)을 이해한다고 할 수 있다.

2.7 XML/EDI 시스템 기술요소

XML/EDI 시스템을 구성하는 기술요소로는 XML, 메시지구조, 템플릿(Templates),

에이전트(Agent), 저장소(Repository)등이 있으며,

- XML은 웹 자료교환 방식으로 이용된다
- 메시지 구조는 상호 협의된 전달방식을 이용한다.
- 템플릿을 이용하여 업무의 진행논리를 포함한다.
- 에이전트를 이용하여 자료처리 기능 등 특별한 기능을 수행하기 위하여 이용한다.
- 저장소는 자료의 유지 관리용으로 존재한다.



<그림 2> XML/EDI 기술의 융합

기술 요소 중 템플릿을 이용함으로써 전통적인 EDI 메시지 (X12, EDIFACT등)와 XML/EDI를 연결할 수 있다. 송수신되는 메시지의 구조정보를 갖고있는 이러한 템플릿을 통하여 XML/EDI는 외부의 변환 지원 없이 송수신 되는 메시지를 재 생성할 수 있게 된다.[1, 6, 8]

2.8 기존 VAN/EDI 시스템과의 비교

XML/EDI를 사용할 경우 기존의 EDI 서비스를 인터넷/웹 플랫폼으로 옮김으로써 사용자들은 서로 다른 EDI 전용 소프트웨어의 사용에 따른 불편함이 없어지고, 통합된 환경으로 EDI 서비스를 사용할 수 있게 된다.

| 구분 | 기존 EDI | XML/EDI |
|---------|--|---|
| 시스템 호환성 | 기존의 시스템을 구축하고자 하는 EDI 시스템에 맞게 변환해야 한다. 즉 각 사용자마다 EDI 변환 소프트웨어를 설치해야 하고, 각 소프트웨어도 제조업체마다 다르다. | 별도의 소프트웨어는 필요 없고, 상호 약속한 통신 방식과 웹브라우저만 준비하면 된다. |
| DB 저장 | 전달된 EDI 문서를 사람이 볼 수 있게 문서화시켜 직접 수작업으로 테이블 형태로 저장한다. | XML 인스턴스의 태그를 파싱해서 테이블 형태로 저장한다. |
| 문서 수신 | 자체 브라우저가 필요하다. | XML을 지원 하는 웹브라우저만 있으면 된다. |
| 프로그램 명 | EDI에 관한 상당한 세부 배경지식이 필요하고, 통신/변환/응용 소프트웨어 모듈을 개발해야 한다. | XML DTD 구조와 전체 프로세스 윤곽에 대한 지식만 보유하고 있으면 되고, 인스턴스를 생성시켜줄 수 있는 프로그램만 개발하면 된다. |

또한 시스템 개발 및 유지보수비가 기존의 복잡하고 확장성이 없는 EDI 전용 시스템보다 저렴해진다. 기존의 EDI 서비스는 대부분 일괄처리형태로 업무를 처리하였지만 XML/EDI를 통하여 일괄처리, 상호대화, CGI FORM, 실시간 문서 교환 기능이 통합된 형태로 제공된다.

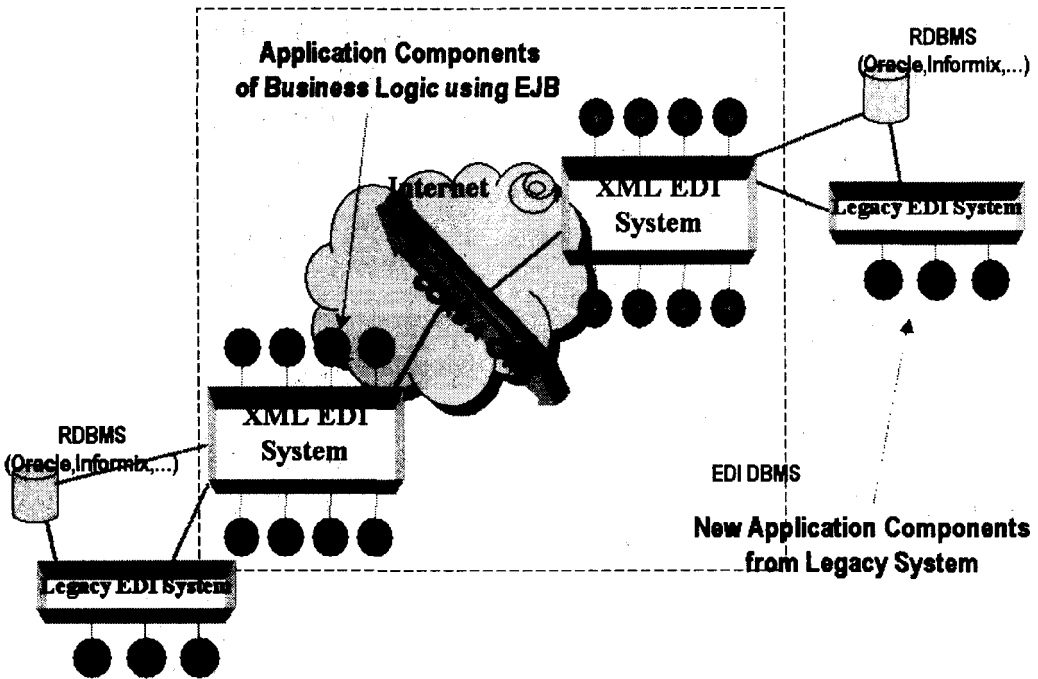
XML의 문서 구조의 정보를 이용하여 더욱 정확하고 효율적인 문서 저장 및 검색이 가능하다. 문서를 구성하는 각각의 요소들을 객체단위로 처리하여 시스템의 확장성을 제공한다. XML/EDI의 활용 효과는 무엇보다도 기존의 EDI 환경의 제약점과 한계성을 해결할 수 있다는 것이 가장 큰 요소로 작용할 것이다.

3. 구현된 XML/EDI 시스템 설계 및 구현

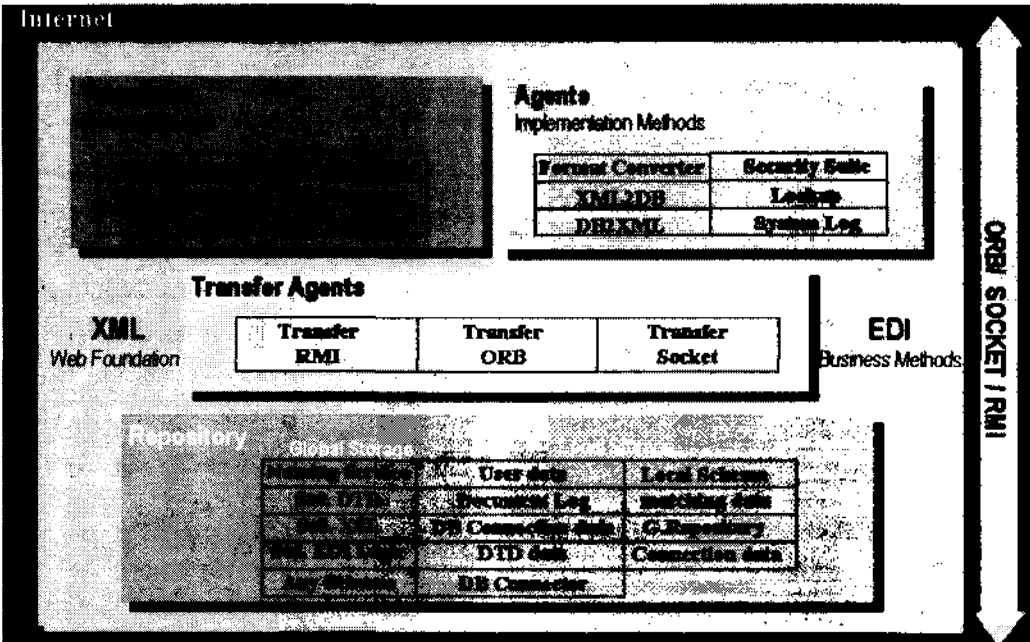
본 절에서 현재 구현하고자 하는 EDI 시스템의 구조와, 모델링, 구현된 컴포넌트에 대해서 기술하겠다.

3.1 XML/EDI 구조도

<그림 3>은 구현한 XML/EDI 시스템의 구조도를 보여준다. 각각의 회사들이 XML/EDI 시스템을 가지고 있는 상태에서 인터넷(소켓이나 RMI, ORB)을 통해서 XML 문서를 주고 받을 수 있으며, 자사의 리가사(legacy) 데이터베이스로 XML 문서의 데이터를 저장할 수 있는 구조를 보여주고 있다.



<그림 3> XML/EDI 시스템 구조도



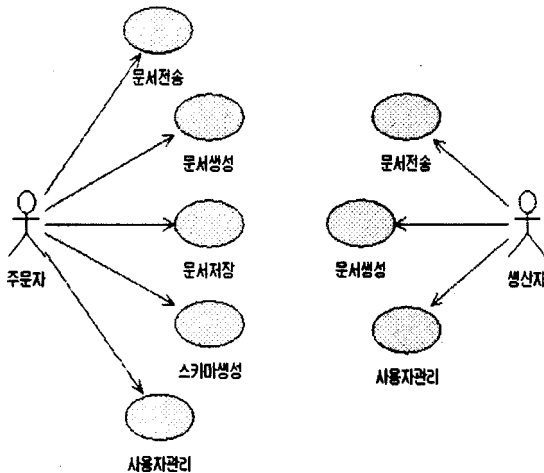
<그림 4> 구현된 XML/EDI 구성 컴포넌트

이러한 XML/EDI 시스템 구조를 세분화하여 <그림 4>와 같이 시스템의 구성도를 그려보았다.

XML 문서 포맷 및 스타일 포맷 등을 담당하는 Template 부분과 데이터베이스로 XML 문서 저장 및 추출을 담당하는 Agent, 외부시스템과 문서 전송 및 수신을 담당하고 있는 Transfer Agent, 시스템 운영에 있어 데이터베이스 연결 등 각종 기능을 담당하는 Repository로 구성된다.

3.2 Use Case 모델링

XML/EDI 시스템에서는 <그림5>와 같이 물품을 주문하는 사람과 주문한 물품을 제공하는 생산자를 액터(Actor)로 정했다. 주문자는 자사에 데이터베이스나 파일의 형태로 문서를 저장할 수 있으며, 생산자는

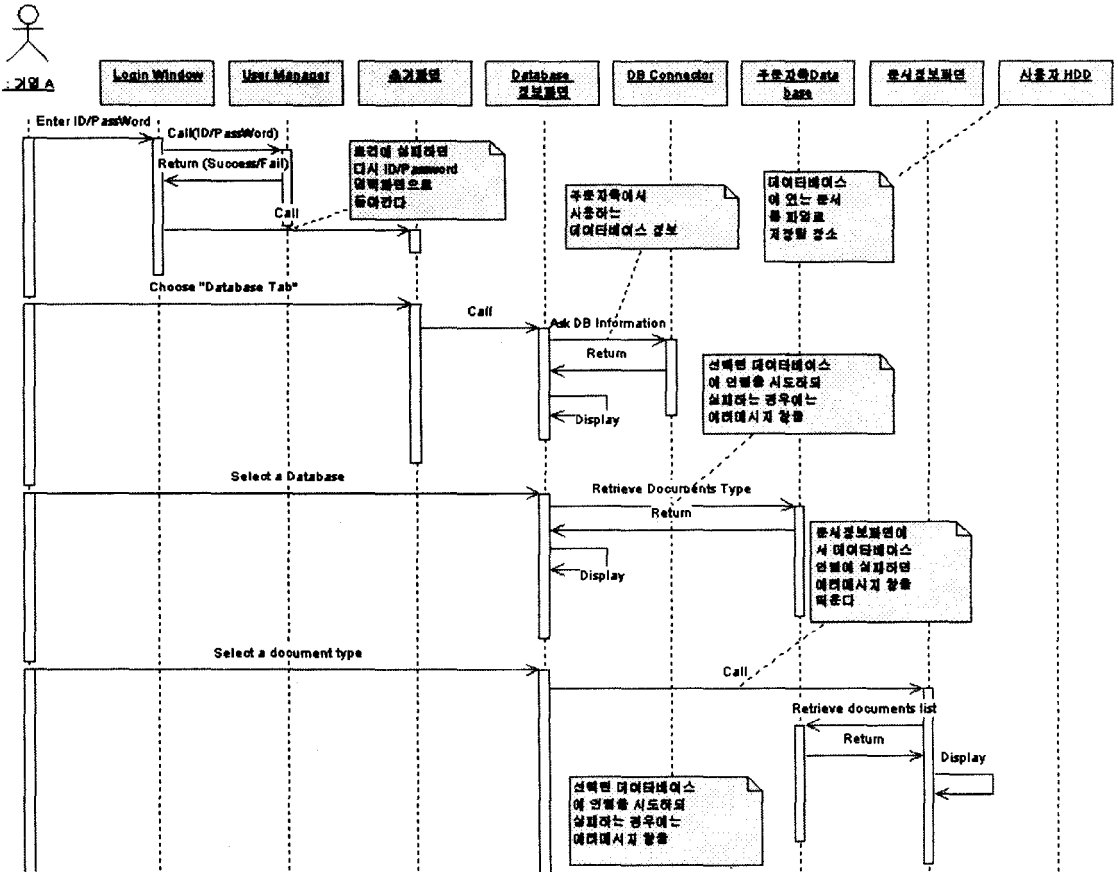


<그림 5> Use case 모델링

시스템의 축소된 형태로 문서를 파일로 보관하는 것으로 보았다. 시스템에서의 주된 기능으로는 Case로 정한 것처럼 문서의 생성과 저장, 전송 등이 있으며, 자사의 데이터베이스와의 연동을 위해서 스키마를 생성하고 시스템 사용자를 관리할 수 있게 하였다.[7]

3.3 동적 모델링

시스템 내부의 객체간의 동적흐름을 나타내는 동적모델링(Sequence Diagram)을 이용하여 위의 Use Case 모델링에서 나타난 각 기능들에 대한 모델링을 하였다. <그림 6>에서는 데이터베이스에 있는 문서를 파일로 바꾸는 기능(DB2XML)을 동적 모델링으로 나타낸 것이다. 우선, DB2XML 기능 흐름을 살펴보면, “사용자가 XML/EDI 시스템에 로그인을 하여, 초기화면에서 불러올 문서를 가지고 있는 데이터베이스와 해당 문서를 선택하면, 데이터베이스에 있는 XML 문서를 사용자 하드디스크에 저장한다”는 시나리오이다. 여기에서 ‘사용자 로그인’과 사용자 인증을 위한 ‘사용자 매니저’, 인증을 통해 사용자가 처음 접하는 ‘시스템 초기화면’, 사용자 측에서 가지고 있는 데이터베이스 연결을 관리하는 ‘DB 연결관리자’, ‘DB 정보화면’, ‘문서정보화면’ 등을 객체로 정했다. 정상적인 흐름 외는 예외상황이나 애러상황으로 메모를 통해 흐름도를 볼 수 있도록 하였다.[7]



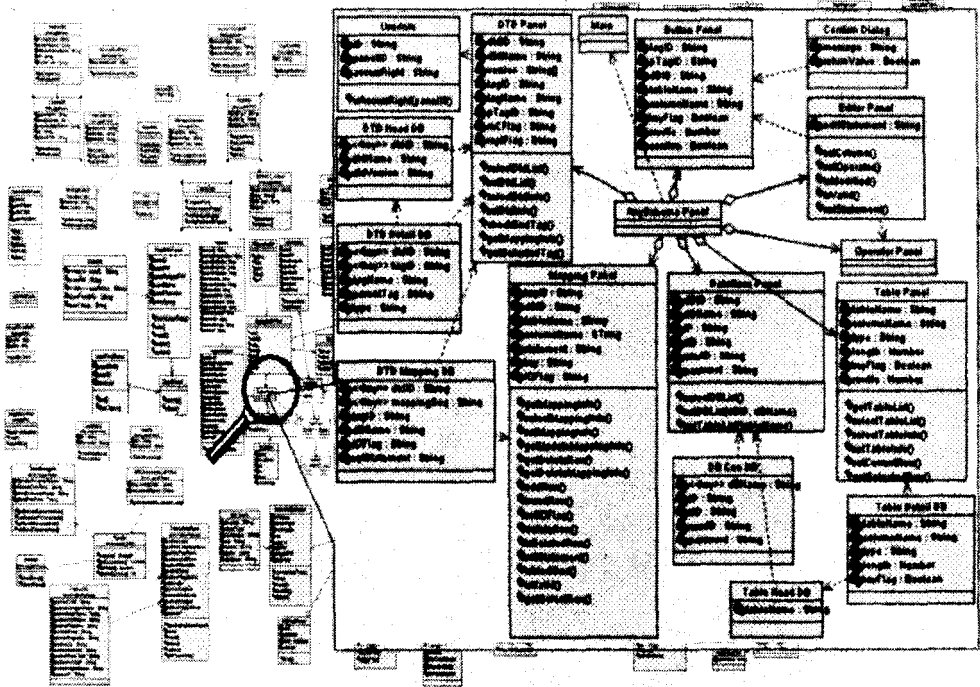
<그림 6> DB2XML의 동적모델

3.4 클래스 모델링

동적 모델링을 마친 후 구현 전단계인 클래스 모델링을 한다. 동적 모델링에서 객체로 설정한 것이 대부분 클래스로 표현되며, 각 기능별로 모델링한 클래스를 통합과정을 거쳐 상속 및 공통 클래스를 도출해낸다.[7]

4. Legacy 데이터베이스와의 연동구현

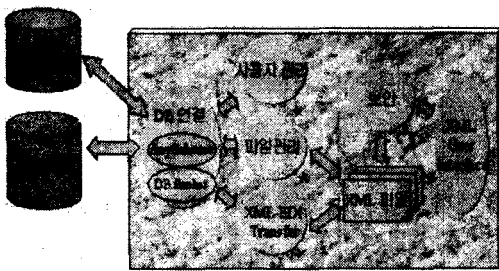
기존의 데이터베이스와의 연동을 위해서 AnySchema라는 컴포넌트를 만들었다. 이 컴포넌트는 XML/EDI 시스템에서 사용할 XML DTD의 변동과 각 기업 내부의 데이터베이스 구조에 따라 유동적으로 문서 데이터를 저장할 수 있도록 한다. 즉, 문서 생성과 저장을 위한 컴포넌트인 'DB2XML'과 'XML2DB'는 XML DTD와 데이터베이스



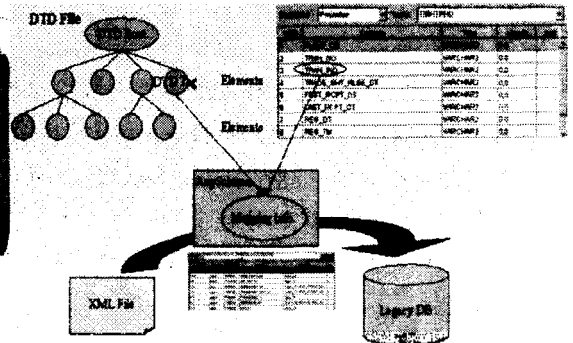
<그림 7> 통합된 클래스 모델링

구조에 따라 유연하게 생성 혹은 저장되어야 하는데, 이런 DTD와 기업의 데이터베이스 스키마와의 동적인 결합 테이블을 만들어 주는 비주얼 데이터 생성 컴포넌트이다. <그림 8>에서는 전체 시스템에서 AnySchema의 위치를 보여준다.

<그림 9>에서는 수신된 XML문서가 Legacy 데이터베이스에 저장되는 플로우를 나타내었다. 수신된 XML 문서는 AnySchema로부터 맵핑정보를 가져와 Legacy 데이터베이스에 맵핑된 테이블 컬럼에 데이터를 저장한다.

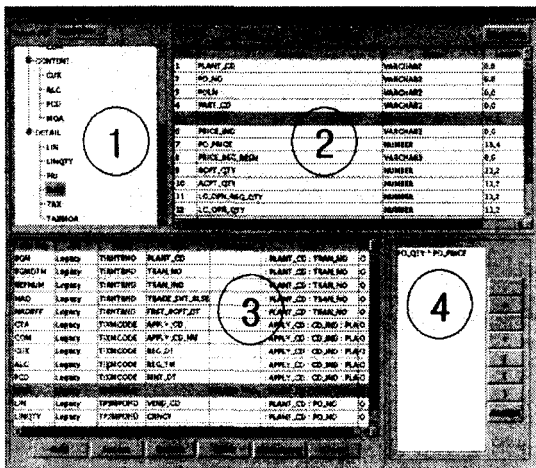


<그림 8> 전체 시스템에서 AnySchema 위치



<그림 9> AnySchema 가능 흐름도

<그림 10>에서는 구현한 AnySchema의 화면을 나타낸 것이다. ①번 창에서 맵핑시킬 DTD의 엘리먼트를 트리로 나타낸다. 이때, DTD에 대한 정보는 이미 데이터베이스에 저장되어 있으며, 이 DTD 정보를 가져오는 것이다. ②번창에서는 ‘Database’의 정보를 나타낸다. 여기에는 연결된 데이터베이스의 테이블과 컬럼에 대한 정보를 보여준다. ①번과 ②번창에서 맵핑시킨 정보는 ③번창에서 생성된다. 이 생성된 정보는 데이터베이스에 맵핑정보만을 가지는 별도의 테이블로 저장된다. 이 테이블은 XML 문서를 수신했을 때, 이 문서의 데이터를 리가시 데이터베이스에 저장할 때 사용하는 테이블이다.



<그림 10> AnySchema 구현화면

이러한 AnySchema는 주고받는 XML 문서의 양이 많은 경우 수신된 XML 문서 데이터를 legacy 데이터베이스에 저장하는데 있어 시스템 부담이 적게 든다. 따라서, 이 기능은 한정된 DTD를 사용하며, 많은

XML 문서를 수신하는 환경에 적합하다.

5. 구현된 시스템 기능 개요

구현한 XML/EDI 시스템은 크게 문서 송수신을 담당하는 “Transfer Agents”와 기능 로직을 담고있는 “Agents”, 프로세싱 로직을 담당하는 “Templates”, XML 문서 및 데이터를 저장하는 “Repository”로 구성되어 있다.

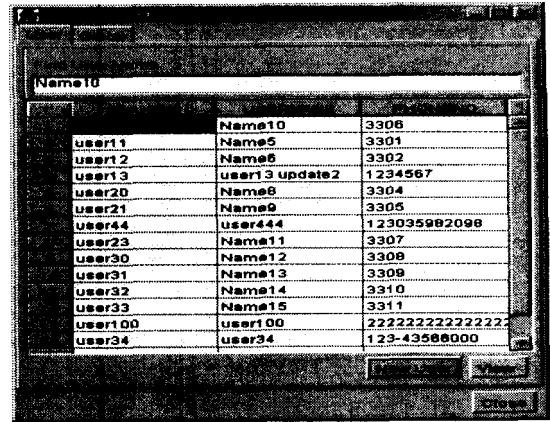
5.1 XML2DB와 DB2XML

XML2DB는 수신된 XML문서를 사용자가 지정한 데이터베이스에 저장하는 기능을 수행한다. 송수신하는 XML문서는 반드시 상호 협의된 DTD에 따라 작성되었으며, 이 DTD는 문서를 주고받는 사용자들 모두 가지고 있는 상태로 가정한다. 시스템 데이터베이스에 저장하려는 문서를 지정된 데이터베이스에 저장하고, 또, 문서의 데이터를 추출하여 맵핑정보에 따라 기존의 리가시(legacy) 데이터베이스에 저장한다. 기존의 리가시 데이터베이스에 데이터를 저장함으로써, 이 데이터베이스의 데이터를 이용하는 응용 프로그램들을 계속 사용할 수 있도록 한다.

DB2XML은 자신의 데이터베이스에 저장된 XML문서 데이터로부터 XML파일 형태로 미리 정해진 경로와 파일명으로 사용자 하드디스크에 저장한다. 사용자는 초기 화면의 데이터베이스 정보 창을 통해 사용자의 데이터베이스에 저장된 XML문서명과 파트너 이름, 관계자명 등을 보면서 XML문서 파일로 전환시킨다.

5.2 Transfer

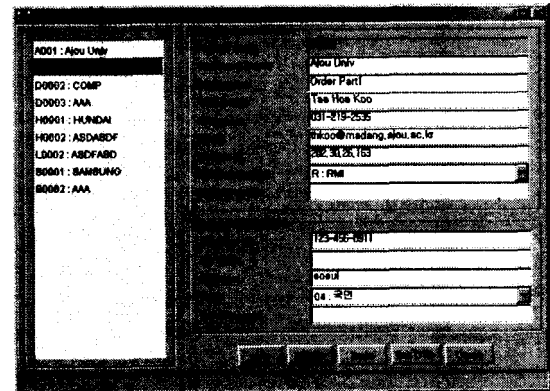
생성된 XML문서를 상대방에게 전송할 수 있도록 해주는 문서 전송 기능을 제공한다. Transfer는 생성된 문서명을 가지고 PartnerManager로부터 수신자의 정보를 알아내며 수신객체의 이름과 포트 정보를 받아 문서를 송신한다. Workflow 설정에 따라 문서 전송이 완료되면 수신자의 Email로 문서 송신 내용을 가진 Email을 발송할 수 있다. 현재 Transfer는 이미 정해진 상대의 IP와 Port정보를 담고 송수신을 하고 있다.



<그림 11> User Manager 화면

5.3 User Manager

XML/EDI시스템 사용자 관리와 사용자 그룹 정보를 관리하는 기능이다. 사용자의 이름과 연락처, Password를 입력하여 새로운 사용자 등록이 가능하다. 그룹설정은 단순히 등록된 사용자들을 그룹별로 나누기 위한 것으로 앞으로 그룹에 따라 사용자가 사용할 수 있는 문서종류에 접근 및 사용제한을 두려고 한다. <그림 11>은 User Manager 기능의 화면이다.



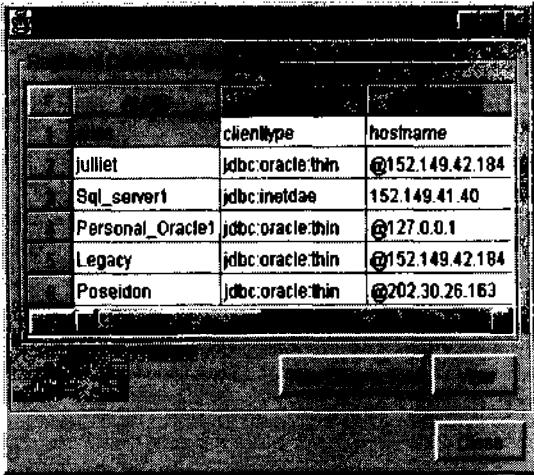
<그림 12> Partner Manager 화면

5.4 Partner Manager

시스템을 이용해서 문서를 주고 받을 상대방에 대한 정보를 관리한다. 상대방 회사명과 부서명, 담당자명, 연락처 등에 관한 정보와 문서 송수신을 위한 IP주소와 통신방식을 설정하여 Transfer기능이 문서 전달을 위한 정보를 갖을 수 있게 한다. <그림 12>는 Partner Manager 화면이다.

5.5 DB Connection 정보관리

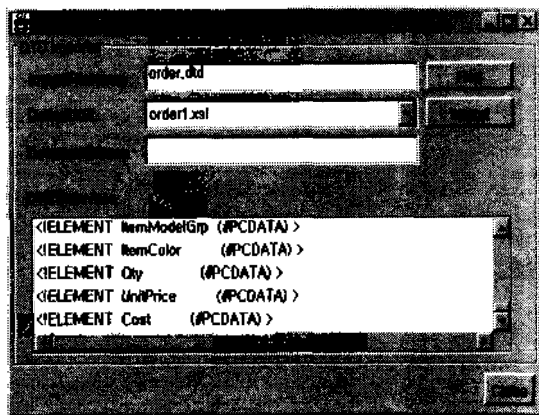
시스템 사용자의 데이터베이스에 관련된 정보를 담고 있다. 등록된 데이터베이스의 가명(Alias)과 시스템과 데이터베이스 간의 연결을 위한 정보(ClientType), 연결을 위한 IP번호, Driver 번호, 사용자명과 Password 정보등을 가지고 사용자가 원하는 데이터베이스에 연결시켜 줄 수 있는 정보를 가지고 있으며, 새로운 데이터베이스를 등록시켜 줄 수 있다.



<그림 13> DB Connection 화면

5.6 DTD Importer

시스템 사용자가 사용하는 DTD와 함께 XML문서의 스타일 정보를 지닌 XSL과 연결을 시켜준다. 즉, 사용자는 자신이 사용하고 있는 DTD를 선택한 후에, 이 DTD를



<그림 14 > DTD Importer 화면

기반으로 작성된 XML문서에 기본 스타일로 적용할 XSL 파일을 설정하는 것이다. <그림 14>와 같이 ImportFileName 항목에서 DTD 파일을 선택하면, Default XSL 항목에서 DTD와 XSL의 맵핑정보를 가지고 있는 테이블로부터 적용 가능한 XSL파일정보를 가져와 리스트로 보여준다.

6. 결론

XML/EDI 시스템은 HTML의 차세대 인터넷 정보표현에 있어 신기술로 평가되어 활발한 응용개발이 이루어지고 있는 XML 기술을 EDI메시지에 적용함으로써 전통적인 EDI 시스템의 여러 가지 문제점을 해결하고자 하는 차세대 EDI에 관한 연구 중의 하나이다. 본 논문에서는 기존의 VAN을 이용하는 EDI 시스템에서 웹을 기반으로 한 XML/EDI 시스템을 객체지향 방식으로 분석 및 설계, 구현함으로써 기존의 EDI 시스템보다 비용측면에서 효율적으로 도입 및 운영할 수 있다. 또, 기존의 데이터베이스와도 연동할 수 있게끔 XML DTD와 데이터베이스 스키마를 동적으로 결합시켜주는 AnySchema를 구현하였다.

추후로는 XML 문서의 데이터를 효율적으로 데이터베이스에 저장하여 기업의 다른 어플리케이션에서도 직접 데이터베이스에 연결하여 데이터를 가공할 수 있는 연구가 필요하다.

참고문헌

- [1] 강재구, "B-to-B 전자상거래 시스템을 위한 XML/EDI 구축방안에 대한 연구", 세종대학교 정보통신대학원
- [2] 김은, "EDI 공통교재", 한국전산원
- [3] 이창호, "새로운 정보교환 패러다임: XML/EDI", EDI/EC 매거진 1998 가을호(12~17p)
- [4] 정희경, "XML 가이드", 서울그린출판사, 1998
- [5] Alex Ceponkus, Faraz Hoodbhoy, Alex Ceponkas John, "Applied XML: A Toolkit for Programmers" Wiley & Sons, 1999
- [6] Bruce Peat & David Webber, "Introducing XML/EDI",
<http://www.xmledi-group.org/xmledigroup/start.htm>
- [7] Hans-Erik Eriksson and Magnus Penker, "UML Toolkit", Wiley, 1998
- [8] Martin Bryan, "Guidelines for using XML for Electronic Data Interchange",
<http://www.geocities.com/WallStreet/Floor/5815/guide.htm>
- [9] Richard Anderson, Mark Birbeck, Michael Kay etc, "Professional XML", Wrox, 2000
- [10] Microsoft, "XML Developer's guide", <http://msdn.microsoft.com/xml/default.asp>
- [11] Microsoft, "XSL Developer's guide", <http://msdn.microsoft.com/xml/default.asp>
- [12] W3C, "Extensible Markup Language 1.0", <http://www.w3c.org/TR/1998/REC-xml-19980210.html>, Feb. 1998.

저자소개

구태희 (e-mail : thkoo@madang.ajou.ac.kr)

1999년 아주대 컴퓨터 공학 학사
 2001년 아주대 컴퓨터 공학 석사
 2001년 ~ 현재 (주)COTECH 사 근무
 관심분야 : 데이터베이스 시스템, XML

변광준 (e-mail : byeon@madang.ajou.ac.kr)

1985년 서울대 컴퓨터 공학 학사
 1987년 Pennsylvania of State University 전산학 석사
 1993년 University of Southern California 전산학 박사
 1994년 ~ 현재 아주대 정보 및 컴퓨터공학부 조교수, 부교수
 관심분야 : 멀티미디어 데이터베이스, 분산 데이터베이스 시스템, 분산 객체 컴퓨팅

황인준 (e-mail : ehwang@madang.ajou.ac.kr)

1988년 서울대 컴퓨터 공학 학사
 1990년 서울대 컴퓨터 공학 석사
 1998년 University of Maryland at College Park 전산학 박사
 1998년 ~ 1999년 미국 보우어 주립대학 조교수
 1998년 ~ 1999년 HRL Research Lab Visiting Professor
 1999년 ~ 현재 아주대 정보 및 컴퓨터 공학부 조교수
 관심분야 : 분산 VOD 시스템, 멀티미디어 데이터베이스,