

인트라넷 환경하에서 문서형식 서버를 갖는 전자문서교환 시스템에 관한 연구

장태우* · 성제훈* · 서용원* · 함주호*

A Study on Electric Data Interchange System with Format Server under Intranet Environments

Tae-Woo Chang · Che-Hoon Sung · Yong-Won Seo · Ju-Ho Hahm

〈Abstract〉

Among the driving methods of CALS, Electronic Data Interchange (EDI), the pivot of procurement and logistics, is becoming the standard of data interchange. In this article, we examine the standard, growth, and status of EDI, and present a scheme to make it available via Internet that grow to be a standardized network, in comparison with a method to use a Value Added Network(VAN). So we studied an implementation of EDI system to exchange and process the business intra-works in order to integrate it into Intranet environment. As well, we present a client/server model running under Intranet environment, and a system with a format server to share the load of server and client. To do this, we list the functions and requirements of EDI system, connect a database to the Intranet, and develop applications using Java™.

1. 서 론

기업간의 경쟁이 치열해지는 환경에서 업무의 흐름과 각종 상거래에서 시간 요소가 중요하게 인식되어감에 따라 업무의 주기가 짧아지고 있으며, 신속한 거래가 요구됨에 따라 필요한 정보를 효과적으로, 그리고 각 부문별 정보를 통합적으로 이용할 수 있는 정보기술이 도입되었다.

이러한 정보기술중, 미국방부에서 군수지원 업무를 위해 시작된 CALS(Computer Aided Logistics Support)는 기업의 상거래까지 확장된 현재의 CALS(Commerce At Light Speed)로 변화발전하였으며, CALS의 추진 방식들중에서 조달과 물류부문의 전자문서교환(Electric Data Interchange:EDI)은 상거래 부분의 표준이 되고 있다.

EDI분야에 있어서는 국내에서는 한국EDIFACT위원회가 중

심이 되어 여러 부가가치망(Value Added Network:VAN) 사업자들이 EDI를 실현하고 있으나 VAN을 통하여 EDI를 이용할 때는 소요비용, 사용불편성 등의 문제가 있으므로 VAN을 이용하지 않는 방법의 개발이 필요하며, 기업내부 정보의 통합적 관리, 기업외부 정보의 효율적 이용을 위해 인터넷을 하부구조로 갖는 인트라넷상에 EDI를 연계시키는 방법이 요구되고 있다.

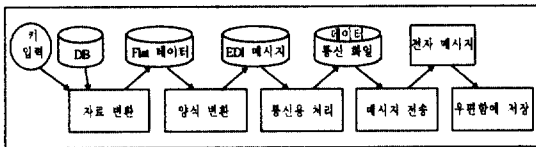
따라서 본연구는 EDI의 운용에 있어서 VAN을 통하거나 거래 당사자간에 직접 연결하는 기존의 방법의 대안으로 인터넷 기반 EDI 시스템을 제시하며, EDI표준을 수용하여 문서형식을 지원할 수 있는 방안 및 시스템 프로토타입의 구축과 타 시스템과의 연계운용방안에 대해 연구한다.

2. EDI(전자문서교환)

EDI는 서로 다른 조직 또는 기업간에 약속된 형식을 사용하여 상업적 또는 행정상의 반복적인 거래를 컴퓨터 또는 응용 프로그램 간에 행하는 것을 말한다. 또는 구조화되고 표준화된 형식을 갖춘 문서의 표준과 그 교환방법을 말하기도 한다.

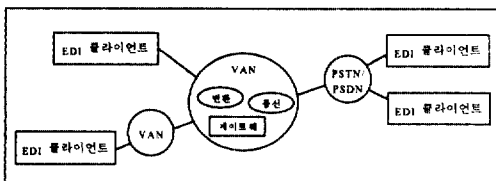
EDI는 정보전달 규약(통신 프로토콜), 정보표현 규약(비즈니스 프로토콜), 업무운용 규약, 거래기본 규약과 같은 규약을 필요로 하며, 현재 통용되는 EDI 표준으로는 크게 ANSI X12와 UN/EDIFACT가 있으나, 현재 EDIFACT로 통합되고 있는 추세이며 국내의 표준도 EDIFACT에 따르므로 본 연구에서는 EDIFACT만을 다룬다.

EDIFACT 표준은 “(서비스 전송항목)-교환단위-기능그룹-메시지-전송항목-자료항목”의 계층적 구조를 가지며, 일반적인 EDI 소프트웨어는 <그림 1>과 같은 단계별 전송과정으로 작동한다.



<그림 1> 단계별 EDI 문서 전송과정

이러한 EDI를 VAN과 같은 기존 통신사업자들의 서비스를 이용하는 경우, EDI 시스템의 구성은 <그림 2>와 같은 구성을 갖는다. 이러한 기존 VAN을 이용한 EDI 시스템의 문제점은 첫째로 방식측면에서는 VAN에 가입하여야 하며, 그 망을 이용시 전송한 후 수신할 때까지 트래킹을 제 3자인 VAN 사업자에게 의존해야 하며, 발생 시점에서 전송과 수신, 저장을



<그림 2> VAN을 이용하는 EDI 시스템

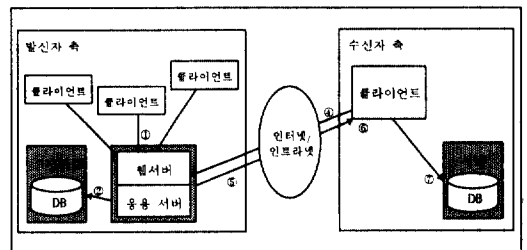
하지 못하고 배치 방식의 저장과 전송만이 가능하고, 그 이용이 복잡하다. 둘째로 비용측면에서는 초기 투자비용은 적게 드나 교환하는 문서량이 많을 경우에는 비용이 많이 증가하게 된다. 하드웨어, 소프트웨어와 통신 부문은 VAN과 호스트 컴퓨터 환경의 주전산기 비용 또는 일반적인 클라이언트/서버 시스템을 사용함으로써 발생하는 항목들이며 그 비용이 크다. 또한 교육 분야 또한 시스템에 중속적인 비용 발생 항목이다.

3. 인터넷과의 EDI의 결합

지금까지 기술한 EDI, 인터넷, 클라이언트/서버 시스템의 개념을 활용한다면 특별한 VAN에 가입할 필요없이 보유하고 있던 자사의 호스트 또는 전용 서버를 이용함으로써 또는 일반 서버를 사용함으로써, VAN을 이용했을 때의 비용을 줄일 수 있다. 3-tier 클라이언트/서버 모형을 사용하여, 기존의 호스트 컴퓨터가 해오던 일을 응용서버와 DB 서버와 같은 여러 서버들이 분담할 수 있고, 또한 인터넷 환경에서는 소프트웨어의 개발과 구현이 일반 클라이언트/서버 시스템에서보다 용이하며, 유지보수 또한 서버쪽에서만 맡게 되어 부담이 적고 교육 비용적인 측면에서도 인터넷에 기반한 EDI 시스템에서는 교육 내용이 일반 브라우저의 사용법 정도에 불과하므로 기존의 방법보다 간단하게 된다.

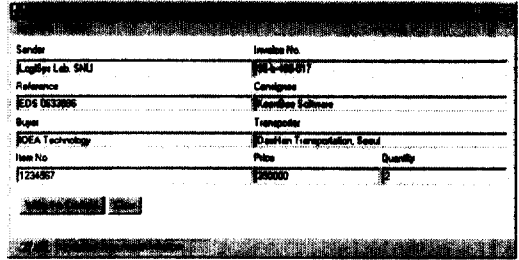
이외에도 비용의 측면에서도 기존 VAN을 사용할 때와는 달리 직접 데이터베이스를 보유하여 이를 다른 업무와 연계하여 이용할 수 있고, 인터넷에서는 데이터베이스가 필수적이며, 이를 이용해 자료의 저장과 검색 등을 효율적으로 할 수 있다.

본 연구에서는 인터넷에 기반하여 EDI 시스템을 구현할 때는 다음의 <그림 3>과 같은 전체적인 구성 및 기능을 갖추도록 시스템을 개발하도록 한다.



<그림 3> 인터넷 기반 EDI 시스템 구성

- ① 브라우저를 이용하여 웹서버에 연결하여 애플릿으로 되어 있는 문서 내용 입력용 응용 프로그램을 호출하여 내용용 입력(웹서버가 있는 컴퓨터는 응용 서버가 됨)
- ② 입력을 완료후, 응용서버는 클라이언트가 되어 DB서버에 연결한 후 내용을 전송
- ③ DB서버는 요청된 데이터를 저장
- ④ 상대방(수신자측) 클라이언트는 브라우저로 HTTP를 통해 발신자 측의 서버에 연결하여 EDI 메시지를 요청
- ⑤ 상대방의 신분 인증 후, 서버는 EDI 메시지를 보냄
- ⑥ 수신된 메시지를 분석하여 HTML문서를 생성, DB서버에 연결하여 내용을 보냄
- ⑦ 받은 데이터를 DB로 저장



〈그림 5〉 송장 입력용 윈도우

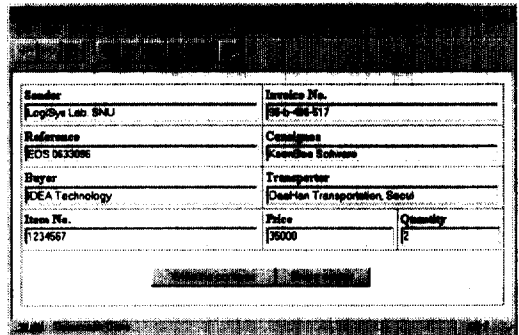
4. 인트라넷 EDI의 일반적 기능 구현

4.1 입력 기능

입력은 전체적으로 인증-설정-작성-전송의 네 단계를 거친다.

먼저 인증단계로 웹 브라우저 상에서 사용자번호와 비밀번호를 입력하면 자료 입력용 윈도우가 생성되도록 한다. 이는 기존의 텍스트 기반에 비해 GUI 기능을 갖도록 하며, 보안상으로도 그 의미가 크다 할 수 있다.

〈그림 4〉는 웹 브라우저 상에서 사용자 인증을 포함하는 로그인 화면을 나타내며, 〈그림 5〉는 송장에 대한 간략화된 입력용 윈도우의 예이다. 플랫폼에 독립적이지 않아도 된다면, Java를 이용하지 않고 C 또는 여타 프로그래밍 언어와 CGI를



〈그림 6〉 CGI와 폼(Form)을 이용한 브라우저 상의 입력화면

이용하여 〈그림 6〉과 같이 브라우저 상에 폼(Form)을 이용하여 입력화면을 설계할 수도 있다.

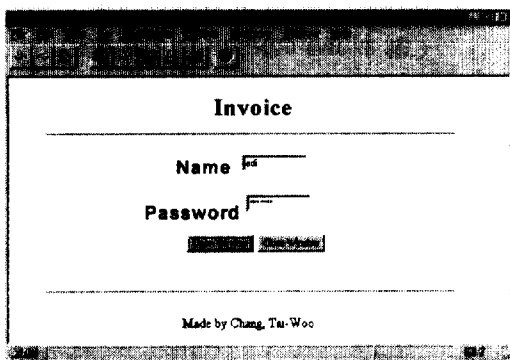
본 설정단계에서는 다음절에서 설명할 사용자 설정 스크립트에서 정의하는 항목만을 입력용 윈도우에 삽입하여 입력하는 작성단계로 넘어가도록 할 것이다. 입력시에 종속적인 자료들은 DB에 의해 참조되어 메시지 생성시 자동으로 삽입되도록 한다. 모든 입력이 끝나면 전송한다.

4.2 양식 변환 기능

양식 변환은 다음에 나올 문서 형식 서버에 대한 기능과 함께 메시지 구조 정의의 스크립트와 밀접한 관계를 가진다. 먼저 일반적인 기능에 대해 언급한 후 내용에 태그를 붙이는 전송 항목의 생성에 대해서는 다음절에서 상세히 설명하기로 한다.

입력받은 자료에 각 전송항목에 해당되는 태그와 종료부호(‘), 그리고 그 사이에 자료항목들을 구분할 수 있는 분리자(+ 또는 :)를 붙이는 EDI메시지로서의 변환기능이 필요하다.

이러한 변환기능에 따라 〈그림 5〉에서 입력한 내용에 대



〈그림 4〉 송장에 대한 브라우저상의 사용자 로그인 화면

```

UNH+ME00001+INVOIC : 90 : 1 : UN'
BGM+95-b-406-517'
DTM+970103:221257'
NAD+SU+LogiSys Lab. SNU'
RFF+SS+EDS 0633096'
NAD+CN+KeenBee Software'
NAD+BY+IDEA Technology'
TDT+DaeHan Transportation, Seoul'
UNS+D'
LIN+1234567'
PRI+AAD+350000'
QTY+1+2'
UNS+S'
MOA+700000'
UNT+15+ME00001'
    
```

〈그림 7〉 입력 내용에 따라 변환된 송장에 대한 EDI 메시지

한 - 간략한 내용만을 포함하는 - EDI 메시지는 〈그림 7〉과 같다.

수신자 측에서는 받은 메시지를 분석하여 태그와 분리자 등을 분리하여 HTML문서의 해당 위치에 삽입하는 기능이 요구된다.

4.3 통신을 위한 기능

웹 서버와 브라우저 간에는 HTTP를 사용하며, 인터넷에서의 사용을 위해 인터넷의 표준 프로토콜인 TCP/IP를 가능케 하도록 해야 하며, 이를 위해 Java 언어의 Socket 클래스와 ServerSocket 클래스를 사용한다. 소켓 클래스를 서버와 클라이언트 양쪽에 각각 생성시키고, 그 소켓을 통해서 데이터를 주고 받는 통신기능을 구현한다.

4.4 보안 기능

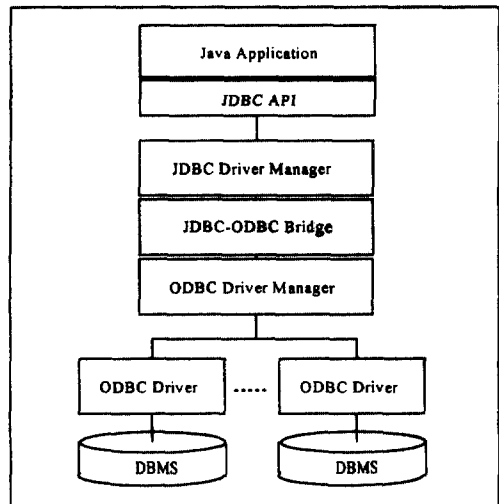
요청을 받을 때의 보안에 대한 기능이 필요하며 이는 인증(Authentication) 절차로써 해결한다. 즉 서버쪽에 특별한 계좌(account)를 주지 않고 응용 프로그램으로써 주어진 ID와 비밀번호를 받아 확인하는 가상계좌(virtual account) 방식으로 한다.

Java 언어는 클라이언트가 서버쪽의 파일에 기록할 수 없는 보안제약을 가지고 있다. 이는 클라이언트 쪽에는 애플릿을, 서버쪽에는 어플리케이션을 설치하여 서로 소켓을 통해 자료를 주고 받은 후 서버쪽 어플리케이션이 파일에 기록하는 방

법으로 해결한다. 이때 서버쪽 어플리케이션이 응용 서버가 되며, 자료를 데이터베이스에 저장할 경우 DB서버와 함께 3-tier 클라이언트/서버 시스템(DB서버와 응용서버가 분리된 형태의 클라이언트/서버 시스템)을 이룬다.

4.5 DB와의 연계방안

인터넷의 구현을 위해 Java를 이용하여 응용 프로그램을 구성하므로 그에 가장 적합한 데이터베이스를 사용하기 위해 많은 DBMS를 Java와 연결가능하게 해주며, DBMS에 따라 문서나 그림 등에 대한 Query를 수용할 수 있어 멀티미디어 자료 등에도 유연성을 가지는 JDBC(Java DataBase Connectivity)를 3-tier 구조의 미들웨어로 정한다. Java 어플리케이션이 데이터베이스에 접근하는 〈그림 8〉에 나타나 있다.



〈그림 8〉 Java 어플리케이션의 접근 과정

지금까지 기술한 내용은 서버쪽 응용프로그램(Application)만의 진행이었으며, 클라이언트는 서버에 접근하기 위해서 애플릿(Applet)을 사용한다. 클라이언트는 웹 브라우저상에서 애플릿을 통해 입력용 윈도우를 생성시키고 여기에서 입력받은 자료를 서버에 넘겨주며 서버쪽의 어플리케이션이 그 자료를 데이터베이스에 저장할 수 있도록 위와 같은 과정을 거친다.

5. 문서 형식 서버의 구축

5.1 메시지 구조 정의 스크립트

EDIFACT의 문서에 따른 권고안의 메시지 구조에 따라 메시지 구조 정의 스크립트를 만든다. 메시지 구조는 “전송항목 섹션-전송항목 그룹-전송항목”의 계층적 구조를 가지며, 전송항목 섹션은 크게 전체 메시지에 관련된 전송항목들로 구성되

```

HEADER SECTION (
  UNH (M, 1)
  BGM (M, 1)
  DTM (M, 35)
  .....
  SG1 (C, 9) {
    RFF (M, 1)
    DTM (C, 5)
  }
  SG2 (C, 99) {
    NAD (M, 1)
    LOC (C, 25)
    FII (C, 5)
    SG3 (C, 9999) {
      RFF (M, 1)
      DTM (C, 5)
    }
    .....
    SG5 (C, 5) {
      CTA (M, 1)
      COM (C, 5)
    }
  }
  SG6 (C, 5) {
    TAX (M, 1)
    MOA (C, 1)
    LOC (C, 5)
  }
  SG7 (C, 5) {
    CUX (M, 1)
    DTM (C, 5)
  }
  .....
)
DETAIL SECTION (
  .....
)
SUMMARY SECTION (
  UNS (M, 1)
  CNT (C, 10)
  SG48 (M, 100) {
    MOA (M, 1)
  }
  .....
  UNT (M, 1)
)
    
```

〈그림 9〉 송장의 메시지구조 정의 스크립트

는 머리말 부분(Header Section), 상세 부분(Detail Section), 요약 부분(Summary Section)으로 구성된다. 전송항목 그룹은 2개 이상의 전송항목을 포함하며 다른 전송항목 그룹을 하위 계층으로 포함하기도 한다. 각 전송항목은 영문 대문자 3자로 된 태그, 태그의 이름, 상태(필수, 조건), 허용 반복 횟수로 구성된다.

메시지의 구조가 위와 같은 계층적 구조를 가지므로 스크립트 역시 계층적으로 작성되며, 전송항목의 반복 횟수는 허용한도 내에서 ‘6.2 사용자 설정 스크립트’의 결정에 따라 입력 문서 형식에 대한 포함 여부를 결정한다.

〈그림 9〉는 표준으로 확정된 송장(UNSM Invoice)의 메시지 구조에 따라 작성된 메시지 구조 정의 스크립트의 예이다. 전송항목의 순서에 따라 머리말 부분, 상세 부분, 요약 부분으로 나뉘며, 중괄호는 전송항목 그룹 간의 포함관계를, 괄호 안은 전송항목의 상태와 허용 반복횟수를 나타낸다.

5.2 사용자 설정 스크립트

사용자 설정 스크립트는 메시지 구조 정의 스크립트를 참고로 하여 문서 또는 메시지 작성시에 사용자에게 필요한 전송항목들과 반복적인 전송항목의 반복 횟수를 입력받아 입력용 애플릿을 만드는데 기초로 삼기 위해 생성시키는 스크립트이다.

EDIFACT에 제시된 표준 메시지 구조에 나타나는 모든 전송항목을 포함하는 문서는 없으므로 사용자에게 전체적인 메시지 구조를 제시한 후 필요한 항목을 선택하게 한다. 이때 필수적인(Mandatory) 전송항목은 반드시 포함되도록 한다.

EDI 메시지에 입력할 내용들 중 기능적으로 연관된 특정 전송항목들은 같은 태그를 갖는 전송항목의 반복횟수가 가변적이므로, 사용자에게 먼저 그 횟수를 입력 받은 후 애플릿을 그에 따라 생성시킬 필요가 있다. 이를 해결하기 위해 EDIFACT 권고안에 따른 메시지들의 전송항목 그룹 중 반복되는 사항들

입력할 품목 수는?	SG25 (1)
	LIN (3)
	QTY (3)
	SG26
	MOA (1)
3

〈그림 10〉 송장의 사용자 설정 스크립트

을 뽑아 사용자 설정 스크립트를 먼저 정의해 놓으며, 이를 참고하여 입력용 윈도우가 생성되도록 한다. <그림 10>은 사용자 설정 스크립트의 예이다.

5.3 입력 문서의 형식 생성

메시지 구조 스크립트에서 전송항목의 상태와 반복횟수가 결정되면 모든 문서 내용 입력 프로그램(애플릿)들은 이 계층적 구조를 갖는 스크립트 형식에 따라 결정된 내용을 갖는, 위에서 설명한 사용자 설정 스크립트와 함께 참조되어 만들어져야 한다.

지금까지 설명한 발신자 측에서 서버가 하는 작업과 그 과정을 나타내면 다음과 같다.

- 스크립트를 참조하여 내용을 읽어들이
- 정해진 포트에 소켓을 생성
- 클라이언트로부터 내용을 받아들이
- 스크립트 내용 참조: 메시지 생성, 저장
- 데이터베이스에 저장

5.4 Parsing과 HTML 문서 생성

메시지를 수신하려는 쪽에서는 서버 쪽 사이트에 웹 브라우저를 써서 HTTP를 통해 접속하고 응용 프로그램(애플릿)으로 소켓을 이용하여 메시지를 읽어온다. 먼저 수신한 EDI 메시지를 분석하여 태그와 실제 데이터를 분리(Parsing)하여야 한다. 분리한 메시지를 HTML 태그들 사이의 적당한 위치에 삽입하여 HTML 문서로 변형시켜 일반 문서와 비슷하도록 다른 페이지로 출력하도록 한다. 다른 용도를 위해서 분리했던 내용 자료들을 데이터베이스에 저장한다.

6. 기존 방식의 EDI와 비교

기존의 EDI 시스템과 비교시 인트라넷 EDI의 장점은 다음과 같다.

- VAN에 가입할 필요가 없어 가입 및 유지 비용을 줄일 수 있다.
- 비싼 호스트 형의 주전산기를 도입할 필요가 없으며, 간단한 서버로 충분하다.

- 소프트웨어 개발이 용이하며 유지보수 를 서버쪽에서 맡게 되어 부담이 적다.
- 시스템 교육이 간단해진다.
- 인트라넷의 데이터베이스와 공유하여 정보의 저장, 검색, 공유가 가능하다.
- 플랫폼에 독립적인 시스템을 갖출 수 있다.
- 배치 방식이 아닌 발생 시점의 전송과 수신, 저장이 가능하다.

하지만 인트라넷을 이용한 EDI의 경우 전문 서비스 업체가 제공하는 VAN에서의 다양한 서비스를 구현하기는 어려운 점과 시스템을 직접 유지해야 한다는 단점이 존재한다.

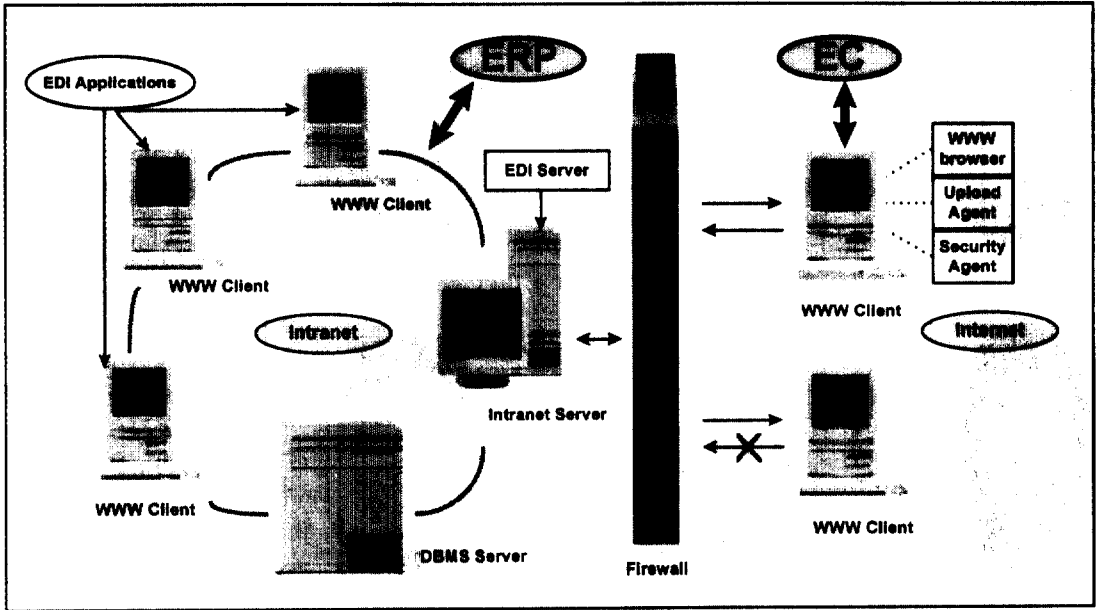
기존의 인트라넷과 비교하면, CGI 위주로 클라이언트 쪽에서는 입력만을 하고 서버 쪽에서 거의 모든 일을 처리 thin client의 일반적인 인트라넷에 비해서, 본 연구에서 제시하는 시스템은 Java 언어를 사용하여 응용을 위한 애플릿을 받아 클라이언트가 일부의 작업을 처리함으로써 전체 시스템의 부담을 분담하는 fat client의 개념이다. 즉, 클라이언트 쪽에서 EDI 메시지의 생성까지를, 서버 쪽에서는 DB 연결과 저장, 그리고 통신을 하도록 한다.

또한 전자우편과 FTP를 사용하는 EDI 시스템에 비해 GUI를 이용한 사용자에게 편의성, 전자우편에는 메시지를 암호화하는 PGP와 PEM의 보안 방법사용의 EDI 메시지에 적용이 가능하다. 전자우편을 이용하면 메시지의 발송 후 수신 확인 방법이 없다는 단점을 가지고, 메시지의 전송을 FTP를 통해 이용하는 방법은 발신자 측의 서버에 수신자 측의 Account를 만들어 주어야 하므로 보안과 관리적인 측면에서 좋지 않은 방법이다. 본 연구에서 제시한 응용 프로그램으로써의 인증확인 절차로 이런 문제를 해결할 수 있다.

7. 결 론

기존에 다른 시스템과 독립적으로 운영했던 EDI 시스템을 인트라넷 환경 하에서 데이터베이스와 연동하여 사용함으로써 전체 기업정보시스템으로의 통합 및 인트라넷의 하부구조인 인터넷에 기반하여 인터넷을 통한 전자상거래(Electronic Commerce:EC)에도 EDI 시스템이 조직 내부 뿐만 아니라 외부까지 연결, 적용이 가능하다.

다음장의 <그림 11>에서 보듯이, 인트라넷 EDI서버를 활용하여 인트라넷 상에서 형식변환 기능을 갖출 수 있다. 또한 인



〈그림 11〉인트라넷 시스템 구성도

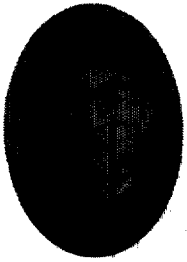
터넷 상의 EC 또는 인트라넷상에서 개발된 전자적 자원 관리 시스템(Enterprise resource Planning:ERP) 상의 EDI 형식의 문서변환을 지원할 수 있다.

이러한 시스템은 만약 표준이 변경된다 하더라도 유연하게 대처할 수 있어, 소프트웨어 측면에서 유지 보수에서의 비용 절감의 효과를 얻을 수 있으며, Java를 사용하여 EDI용 응용 프로그램 자체를 전송받아 사용하는 방법은 다른 인트라넷 모델들에도 적용가능하며, 서버와 클라이언트의 부담을 분담할 수 있다. 또한 기존의 VAN 사업자들을 이용하지 않아도 된다는 장점이 있다.

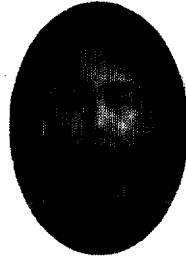
【참고 문헌】

[1] Baker, Richard H., "EDI: What Managers Need to Know About the Revolution in Business Communications", TAB Professional and Reference Books, 1991
 [2] Borenstein, N. and Freed, N., "MIME (Multipurpose

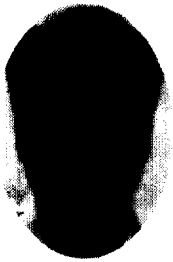
Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993
 [3] UN/EDIFACT Rapporteurs, "United Nations Trade Data Interchange Directory (UNTDID) D.96B", United Nations, 1996
 [4] Hendry, M., "Implementing EDI", Artech House, Inc., 1993
 [5] Houser, W. et al., "EDI Meets the Internet", RFC1865, C. Hage Associates, January 1996
 [6] Sun Microsystems, "The Java™ Language: An Overview", Java white papers, 1995
 [7] 박주석, 김찬수, "인터넷, 인트라넷, 데이터웨어하우징을 통합하는 기업정보시스템의 구축", 한국경영과학회 추계학술대회 발표논문집, 1996
 [8] 최상모, 장병만, "우리나라 무역, 운송분야 EDI System 도입방안에 관한 연구", 경영과학, 제12권 제 1호, 1995



장태우
 1995년 서울대학교 산업공학 학사
 1997년 서울대학교 산업공학 석사
 현재 서울대학교 공과대학 산업공학과 박사과정
 관심분야 internet/intranet 응용 및 구현 Business Process Reengineering



서용원
 1994년 서울대학교 산업공학 학사
 1996년 서울대학교 산업공학 석사
 현재 서울대학교 공과대학 산업공학과 박사과정
 관심분야 internet/intranet 응용프로그램 개발 Enterprise Resource Planning



성재훈
 1994년 서울대학교 산업공학 학사
 1996년 서울대학교 산업공학 석사
 현재 서울대학교 공과대학 산업공학과 박사과정
 관심분야 Supply Chain Management Enterprise Resource Planning



함주호
 1983년 서울대학교 산업공학 학사
 1985년 서울대학교 산업공학 석사
 1990년 Univ of Michigan at Ann Arbor 산업공학 박사
 1990~92년 서울대학교 공학 연구소 연구원
 현재 서울대학교 공과대학 산업공학과 교수
 관심분야 물류관리, 재고관리 Enterprise Resource Planning Business Process Reengineering