
XML 기반의 EDI 문서 처리 시스템 설계 및 구현

임대명 · 정상호 · 류지웅 · 정회경

배재대학교 컴퓨터공학과

The Design and Implementation of EDI document processing system based on XML

Dae-Myung Im · Sang-Ho Cheong · Jee-Woong Ryu · Hoe-Kyung Jung

Dept. of Computer Engineering, Paichai University

E-mail : (carta2513, tophojs, jwryu, hkjung)@pcu.ac.kr

요 약

현재 초고속 인터넷망을 이용한 정보 처리(Information processing) 및 정보 교환(Information exchange)에 대한 연구 및 개발이 활발하게 이루어지고 있다. 이를 위해서는 전자 문서에 대한 좀더 효율적이고 정확한 표준화 작업이 진행되어야 한다. 이에 따라 세계 각국의 표준화 기구들은 각종 표준을 제시하고 있으며, 이중 대표적으로 기업 간에 데이터를 교환하는 표준화 시스템으로 EDI(Electronic Data Interchange)를 들 수 있다. EDI는 전자상거래의 한 형태이며, 기업 간 거래에 관한 데이터와 문서를 표준화를 거쳐 컴퓨터 통신망을 이용하여 거래 당사자 간에 직접 송수신이 이루어진다. 하지만, 기업과 기업이 서로 정의된 방식으로 문서를 전달하는 것이므로 데이터 자체가 고정되어 있고 정형화되어 있어 확장 시에 문제가 발생하고, 구조적인 데이터 검색 및 처리가 불가능하다.

이에 본 논문에서는 구조적이고 확장성이 뛰어난 인터넷 표준 언어인 XML(eXtensible Markup Language)을 이용하여 EDI 문서를 처리하는 시스템을 설계 및 구현하였다.

ABSTRACT

Now, Using high-speed Internet network for information processing and exchange of information on the research and development is active. To do this, Electronic document standards for more efficient and accurate operations shall be conducted. Accordingly, Standards bodies from around the world proposed various standards, and Typically The EDI is standardized system for exchanging data between companies. EDI is one form of e-commerce. And, through the transactions between companies to standardize data and documents. And using a computer network between the parties directly for sending and receiving. But, Companies and other businesses to deliver the document as defined, so data are fixed and formalized at the time of the extension causes problems and structured data search and processing is impossible.

In this paper, While using structure and expanding Internet-standard language the XML for EDI document processing system was designed and implemented.

키워드

XML, EDI, KEDIFACT, DTD, 전자상거래

1. 서 론

국내외를 막론하고 초고속정보통신망의 구축 작업이 활발히 진행되고 있으며, 이에 따라 정보통신망을 이용하여 정보를 교환하고 업무의 효율화를 꾀하려는 노력이 점차 활성화되고 있다. 이

러한 노력이 결실을 맺기 위해서는 우선 전자문서의 표준화 작업이 필요하며 각 표준기구들이 제시한 전자문서에 대한 표준들이 꾸준히 제시되고 있다. 현재 쓰이고 있는 대표적인 형태가 바로 EDI이다[1]. EDI는 기업과 기업이 서로 정의된 방식으로 문서를 전달하는 것이므로 데이터 자체

가 고정되어 있고 정형화 되어 있어서 확장 시에 문제점과 구조적인 데이터 검색 및 처리에 어려움이 있었다. 또한 이 두 기업 간에 데이터를 전달하기 위해 VAN(Value-Added Network)을 이용하여 데이터를 전달했으나 VAN의 폐쇄성으로 인해 확장에 문제가 제시되었다.

이에 본 논문에서는 VAN을 이용하지 인터넷을 기반으로 구조적이고 확장성이 뛰어난 XML을 이용하여 EDI문서를 처리하기 위한 시스템을 설계 및 구현하였다. XML로 기술된 구조화된 전자문서를 이용하여 EDI문서 처리 시스템을 설계 및 구현하였다.

II. 관련연구

2.1 XML

XML은 1996년 W3C(World Wide Web Consortium)에서 제안하여 1998년에 표준으로 제정된 것으로, 웹상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이다[1].

XML의 장점은 XSL(eXtensible Stylesheet Language)을 이용하여 문서로서의 역할을 가지며 문서 그 자체가 데이터로서의 역할을 동시에 가진다는 것이다. 즉 XML은 구조화된 구조를 갖는 자기 서술적(Self Describe)인 언어이기 때문에 응용에서 순수한 데이터로서 사용될 수 있으며, 또한 구조화 되어 있기 때문에 효과적인 정보검색이 가능하다.

2.2 EDI

EDI는 정형화된 거래, 행정 관련 정보를 데이터 기술상의 일정한 통신표준에 입각하여 컴퓨터와 컴퓨터 간에 구조화된 전자 표준 문서나 파일 형태로 교환, 축적, 처리하는 방식이다. 서류 문서를 전자 파일의 한 형태로 생성하여, 종래의 우편, 전화, 인편에 의한 전달 방식 대신 전자적인 방식으로 전송하는 것이라고 규정할 수 있다[2]. EDI 표준은 EDI 사용자간에 교환되는 전자문서의 내용과 구조, 통신 방법 등에 관한 일련의 규칙 및 지침을 말한다[3].

2.3 UN/EDIFACT

UN/EDIFACT란 행정, 무역 및 운송을 위한 전자문서교환에 관한 국제연합규칙이다. 이것은 유럽경제위원회(UN/ECE)에 의해 UNTDID(UN Trade Data Interchange Directory)로 승인, 공표되었으며, 합의된 절차에 따라 유지, 개선되고 있다[3]. 교환되는 EDI메시지는 일정형식의 구조를 가진다. 실제 전달하려는 사용자 자료전송항목을 중심으로 7개의 보조전송항목이 쌍으로 묶여진 구조를 갖게 된다.

III. 시스템 설계

3.1 EDI문서 처리 시스템의 구성

본 논문에서 개발한 EDI문서 처리 시스템은 업무 중 발생하는 문서를 XML로 생성하기 위한 트랜잭션 처리기(Transaction Processor)와 이러한 문서를 작성하기 위하여 사용자의 입력을 받을 수 있는 템플릿 관리기(Template Manager), 기존 EDI시스템과의 호환을 위한 변환 처리기로 구성된다. 그림 1은 시스템 상세 구성도이다.

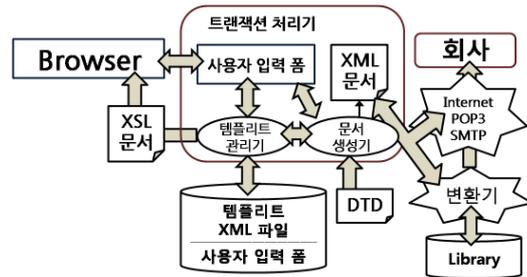


그림 1. 시스템 상세 구성도

본 시스템은 KEDIFACT에 표준 전자 문서에 기반을 두어 개발된 DTD에서 템플릿 파일을 생성하여 저장하고, 사용자가 해당하는 XML문서를 만들기 위해 문서 종류를 선택하면 템플릿 문서와 해당하는 템플릿 입력 폼이 각각 메모리와 브라우저에 표현된다. 표현된 템플릿 입력 폼에 사용자는 데이터 값을 입력하면 그 값을 템플릿 관리기에서 값을 문서 생성기로 넘겨주게 되고 받은 데이터를 이용하여 XML문서로 생성하게 된다. 생성된 문서를 송수신기가 받아 POP3 통신규약에 맞춰 코딩한 후 메일 서버로 송신하게 된다. 만약 생성된 문서가 EDI문서로 변환되어야 한다면 변환기를 통해 변환된 후 송수신기에 보내지게 된다.

3.2 템플릿 관리기

템플릿 관리기는 DTD를 입력 받아 템플릿 XML파일을 생성하고 새로운 DTD에 따라 사용자 입력 폼을 생성할 수 있도록 인터페이스를 제공하며 기존 템플릿 파일과 입력 폼을 관리한다. 사용자가 기존 템플릿 파일을 변경할 때 이에 따라 기존 템플릿 파일을 갱신하고 이 템플릿 파일에 따라 작성된 사용자 입력 폼을 수정한다.

3.3 변환기

변환기는 기존 EDI와의 양립을 위하여 본 시스템에서 제공하는 기능이다. 기존 EDI로 문서를 변환하기 위해서는 EDI메시지를 이루는 세그먼트의 구조를 알고 있어야 한다. 이런 구조를 제공하기 위하여 EDI 세그먼트와 엘리먼트를 XML문서로 표현하여 저장하였다. 이 세그먼트와 엘리먼트

는 문서 변환하는 과정에서 EDI의 구문규칙을 적용하기 위한 정보를 제공한다. 또한 엘리먼트에 각 엘리먼트에 대한 코드와 코드에 대한 설명을 추가하여 저장하였다. 이 엘리먼트 코드는 세그먼트로 정보를 제공하게 되며 변환과정에서 생길 수 있는 예외 처리를 위하여 정보를 제공한다. 아래 그림 2는 변환기의 세부구성도 이다.

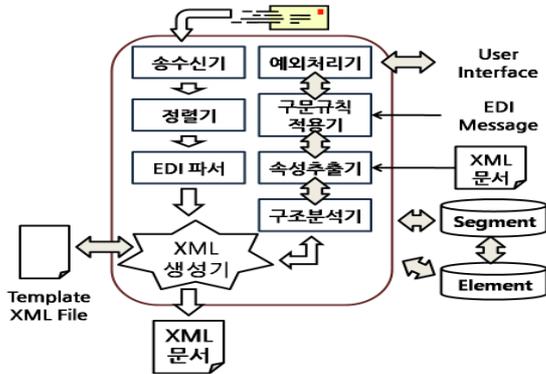


그림 2. 변환기 시스템 세부 구성도

IV. 시스템 구현

4.1 구현환경

본 논문에서 설계 및 구현한 EDI문서 처리 시스템은 Windows XP를 기반으로 작성되었으며 Visual C++ 6.0을 이용하여 구현하였다. XML문서의 검증을 위하여 MSXML파서를 사용하였으며 문서의 브라우징을 위하여 MFC(Microsoft Foundation Class)에서 제공하는 Internet Explorer 컨트롤을 이용하여 브라우징을 처리하였다. DOM 인터페이스를 이용하여 문서 생성 및 템플릿 생성과 관리를 구현 하였으며 문서의 송수신을 위하여 SMTP와 POP3를 이용하여 구현 하였다.

4.2 사용자 인터페이스의 구현

사용자가 편리하게 문서를 만들 수 있도록 사용자가 자신의 작업 창을 구성할 수 있도록 구현하였다. 크게 4가지 작업창으로 구성되었는데 왼쪽 상단의 탐색기 창은 사용자가 수신한 문서와 송신한 문서를 볼 수 있도록 트리 컨트롤로 처리하였으며 생성한 서식문서를 사용자가 이용할 수 있도록 하였다. 엘리먼트 뷰는 XML문서의 구조를 보여준다. 이 창은 서식 문서를 생성하기 위한 인터페이스로도 제공된다. 하단은 파서에서 나오는 에러를 보여주기 위한 에러보고창이다. 에러보고창은 수신 문서나 생성된 문서를 화면에 브라우징 할 때 파서에서 생성되는 에러메시지를 받아 에러보고창으로 보여준다. 그림 3은 사용자 인터페이스 구현 화면이다.

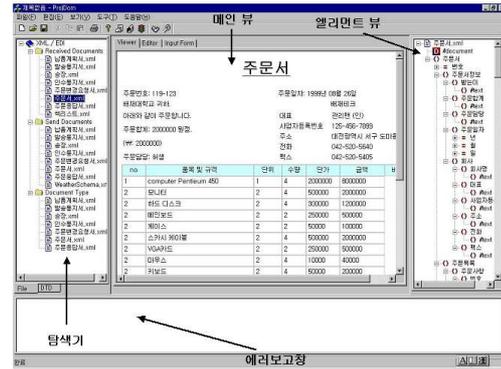


그림 3. 사용자 인터페이스 구현

4.3 템플릿 관리기의 구현

템플릿 관리기는 사용자가 전송하기 원하는 엘리먼트를 선택하여 DTD정보를 포함하는 템플릿 파일을 만들 수 있으며 이 템플릿 파일에 기반을 두어 입력 폼을 생성 및 수정할 수 있도록 구현하였다.

V. 결 론

본 논문은 XML을 이용한 EDI문서 처리 시스템의 설계 및 구현에 관한 것이다. 본 시스템의 장점은 첫째 웹에 기반을 두지 않고 독립 응용프로그램으로 설계 및 구현되었기 때문에 HTTP 통신규약상의 보안 문제를 제거할 수 있으며 둘째 웹에 기반을 두어 고정적인 서비스를 제공 받는 것이 아닌 사용자가 직접 DTD에 따라 문서를 생성 및 관리 할 수 있도록 한 것이 특징이다. 셋째 본 시스템은 XML을 이용하여 EDI문서의 구조정보를 제공함으로써 타 시스템에서의 문서 변환의 방법을 제공하였다.

향후 연구 과제로는 기업의 비즈니스 문서이기 때문에 보안을 위한 시스템이 필요하므로 문서를 암호화 하여 보낼 수 있는 연구가 필요하다. 그리고 현재 기존 EDI와의 맵핑을 위하여 DTD를 사용하게 되는데 이런 DTD를 자동 생성할 수 있는 시스템 개발에 관한 연구가 필요하며 생성된 문서나 수신된 문서의 스타일을 변경 및 생성을 위한 스타일 에디터의 개발이 필요하다.

참고문헌

- [1] 정희경, "XML 가이드", 도서출판 그린, 1999년
- [2] Charles F. Goldfarb, Paul Prescod, "The XML Handbook", Prentice-Hall, 2000년
- [3] Frank Boumphrey의 12명, "XML Applications", Wrox Press, 1998년
- [4] 한국전자거래진흥원, "EDI의 개요", <http://www.kiec.or.kr/>