

XML에 기반한 EDI 문서교환 시스템 설계 및 구현

임 영 태[†] · 한 우 용^{††} · 정 회 경^{†††}

요 약

본 논문에서는 XML(eXtensible Markup Language)을 이용하여 고정적이고 정형화된 EDI(Electronic Data Interchange) 문서를 유동적이고 효율적으로 처리하기 위한 시스템을 설계 및 구현하였다. 본 시스템에서는 사용자가 원하는 문서를 직접 생성할 수 있도록 트랜잭션(Transaction) 처리기와 템플릿(Template) 관리기를 설계 및 구현하였으며, 기존 EDI와의 호환을 위해 변환기를 설계하였다. 또한, 본 시스템의 경우 변환에 반드시 필요한 EDI 메시지 구조를 XML형식으로 저장하고, 이것을 DOM API로 관리하도록 하여 XML의 구조정보를 이용하여 기존 시스템과의 호환성을 고려하였으며, 사용자가 변환기를 이용하여 템플릿 파일을 생성할 수 있고 필요한 구성요소(elements)를 선택 전송할 수 있도록 해당 인터페이스를 제공하도록 하여, 사용자가 전송할 엘리먼트를 선택할 수 있는 인터페이스를 제공하였다. 그리고 아직까지 다른 시스템에서는 문서의 변환에 관한 적절한 해법을 제시하지 못하고 있지만 본 시스템에서는 기존의 시스템이 제공하지 못하는 XML기반의 EDI문서 변환 기법 및 구조정보를 제공하여 그 해법을 제안하고 있다.

Design and Implementation of EDI Document Exchange system based on XML

Young-Tae Im[†] · Woo-Yong Han^{††} · Hoe-Kyung Jung^{†††}

ABSTRACT

This paper presents is for the design and implementation of EDI document exchange system based on XML. To create a customized document of the users' choice, it designed and created the transaction processor and the template manager, and to make it accessible with the original EDI, a converter function is included. Also, on this system, this protocol stores EDI message structure that needed to exchange as XML format and controls it as DOM API for user can use previous system. And provides interface for user can create template files with converter and transfer necessary elements that can be chosen by user. For this purpose, This system proposes a shows structure information and document converting mechanism solution of EDI documents based on by using XML which does not show proper document conversion mechanism solution in other system so far.

1. 서 론

전세계적으로 인터넷의 확산과 더불어 초고속정보통신망의 구축 작업이 활발히 진행되고 있으며, 이에따

라 이러한 정보통신망을 이용하여 정보를 저렴하고 편리하게 교환하며 업무를 효율적으로 처리하기 위한 방안을 모색하고 표준화하려는 노력을 하고 있다. 이러한 노력이 결실을 맺기 위해서는 우선 전자문서의 표준화 작업이 필요하며 각 표준기구들이 제시한 전자문서에 대한 표준들이 꾸준히 제시되고 있다[5, 11]. 이러한 노력의 결과로 가장 일반적으로 쓰이고 있는 대표적인 형태가 바로 EDI(Electronic Data Interchange)이

* 본 논문은 2000년도 충남대학교 RRC 연구과제의 연구비 지원에 의해 연구 되었음.

† 정 회 원 : 배재대학교 대학원 컴퓨터공학과

†† 정 회 원 : 한국전자통신연구원, 컴·소프트웨어 연구소

††† 종신회원 : 배재대학교 컴퓨터공학과 교수

논문접수 : 2000년 9월 20일, 심사완료 : 2000년 12월 29일

다[1]. 현재의 EDI서비스는 송신자, 수신자, VAN(Value-Added Network)간의 교환할 문서를 미리 정의하여 등록시켜 둔 EDI 전용 소프트웨어를 이용하여 전자문서를 주고 받는다[10]. 이러한 환경에서 EDI 서비스는 새로운 형태의 문서를 교환해야 할 필요가 생겼을 경우 사용자마다 새로운 문서에 정보를 등록하여 EDI 전용 소프트웨어의 변경이 필요하다. 그러므로 끊임없이 변화하는 다양한 사용자의 욕구를 만족시키기에는 동적기능이 부족한 것이 EDI 서비스 환경이다.

또한, 이 두 기업간에 데이터를 전달하기 위해 VAN을 이용하여 데이터를 전달했으나 VAN의 폐쇄성으로 인해 확장에 문제가 제시되었다. 이러한 문제점을 해결하기 위하여 다양한 문서구조 표현이 가능한 XML(eXtensible Markup Language)을 활용한 EDI 서비스 환경으로 옮겨가고자 하는 환경변화의 바람이 일고 있다[2,3].

이에 따라 본 논문에서는 폐쇄적인 VAN을 이용하지 않고 기존에 구축되어 있는 인터넷을 기반으로 구조적이고 확장성이 뛰어난 인터넷 표준 언어인 XML을 기반으로 하여 EDI문서를 처리하기 위한 시스템을 설계 및 구현하였다.

본 논문의 연구목적은 XML기반으로 구조화된 전자문서를 이용하여 EDI문서 교환 시스템을 설계하고 구현하는 것이다. 기존 EDI의 문제점인 고정적이고 정형

화된 데이터를 XML로 기술하여 특정 시스템에 종속적이지 않으면서 데이터를 유연하게 확장할 수 있도록 하였으며 이를 위해 XML과 EDI에 관련된 기초 기술 연구 및 EDI 메시지 교환 처리에 관한 연구를 한다.

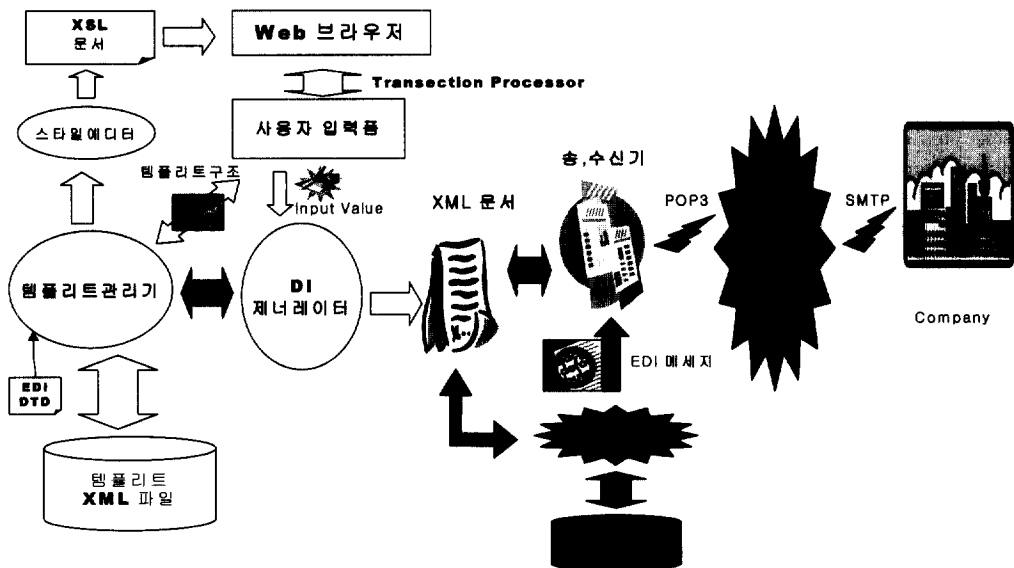
2. EDI문서 교환 시스템 설계

2.1 EDI문서 교환 시스템 구성

본 논문에서 구현된 EDI문서 교환 시스템의 구성은 빈번하게 발생하는 업무 문서를 XML로 생성하기 위한 트랜잭션 처리기와 이러한 문서를 작성하기 위하여 사용자의 입력을 받을 수 있는 템플릿 관리기, 기존 EDI시스템과의 호환을 위한 변환 처리기로 구성되어 있다.

본 시스템은 통신 S/W 계층, 변환 S/W 계층, 응용 S/W 계층의 3가지 계층으로 구성된다. 즉, 응용계층(Application Layer)에서는 사용자 입력을 받아 들이며 이렇게 입력된 데이터를 변환계층(Translation Layer)에서 XML과 EDI문서로 변환하고, 이렇게 변환된 문서를 통신계층(Communication Layer)에서 거래 상대방에게 전송한다. 각 계층들은 다시 상세 부분으로 나뉘게 된다. 아래 (그림 2.1)은 시스템의 상세구성도이다.

본 시스템은 KEDIFACT에 기반하는 표준 전자문서로 개발된 DTD(Document Type Definition)에서 템



(그림 2.1) 시스템 상세 구성도

플리트 파일을 생성하여 저장하고, 사용자가 해당하는 XML문서를 만들기 위해 문서 종류를 선택하면 템플리트 문서와 해당하는 템플리트 입력 폼이 각각 메모리와 브라우저에 표현된다. 표현된 템플리트 입력 폼에 사용자는 데이터값을 입력하면 그 값을 템플리트 관리기에서 값을 문서 생성기로

넘겨주게 되고 받은 데이터를 이용하여 XML문서로 생성하게 된다.

이렇게 생성된 문서를 송,수신기가 받아 POP3 통신 규약에 맞춰 코딩한 다음 메일 서버로 송신하게 된다. 만약 생성된 문서가 EDI문서로 변환 되어야 한다면 변환기(Converter)를 통해 변환된 후 송,수신기에 보내지게 된다.

문서의 스타일정보를 수정하기 위하여 스타일 편집기(Style Editor)가 필요하지만 이것은 향후 연구과제로도 충분히 가치가 있는 것으로 본 논문의 범주에 들지 않으므로 논하지는 않는다.

2.2 트랜잭션 처리기

트랜잭션 처리기에서는 사용자가 사용하려는 문서를 생성시키는 과정을 총괄하여 처리하는 역할을 한다. 이곳에서는 템플리트 관리기에서 생성된 템플리트의 정보와 사용자 입력 폼에서 입력된 데이터를 이용하여 실제 전송하고자 하는 문서를 생성시키는 과정을 수행한다.

사용자는 자신이 전송할 문서를 선택하게 되면 트랜잭션 처리기에서 해당하는 XML템플리트 파일을 메모리로 적재하고, 사용자 입력폼을 화면에 생성하게 된다.

트랜잭션 처리기에서 문서가 생성되는 과정을 나타내 보면 다음과 같다.

1) 전송할 문서 선택

사용자는 전송하고자 하는 템플리트 문서선택

2) 선택된 템플리트 문서 메모리에 적재

선택된 문서에 대해 DOM을 이용하여 메모리로 적재한 후 사용자 입력 폼에 문서정보 제공

3) 사용자 입력

선택된 템플리트 문서에 해당하는 사용자 입력 폼을 화면에 표현한 후 사용자에게 입력 받음.

4) 문서 생성

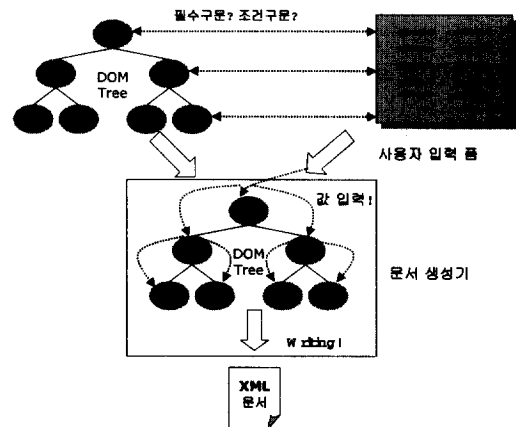
입력된 사용자 데이터를 문서 생성기에서 받아

XML문서 생성.

2.2.1 문서 발생기(DI Generator)

문서 발생기는 DTD에서 엘리먼트 명을 받아 템플리트 XML파일을 생성하며 사용자 입력 템플리트 폼에서 입력된 데이터를 이용하여 XML문서를 만들어 낸다.

(그림 2.2)는 문서 생성기가 XML문서를 생성하는 과정을 나타낸 것으로써 문서 생성기는 사용자 입력값을 DOM(Document Object Model) 트리 각 노드에 DOM API를 이용하여 삽입하고 이것을 파일로 저장함으로써 XML문서를 생성시킨다.



(그림 2.2) 문서생성기

본 논문에서는 사용자가 데이터를 입력할 때 사용자 입력폼에 해당하는 문서의 DTD정보를 가지고 있어야 한다는 것에 착안하여 기존 DTD에서 생성된 XML 템플리트 파일을 DOM을 이용하여 메모리로 적재한 후 사용자 입력 폼에 정보를 제공하는 것으로 설계하였다. 즉, 사용자는 전송하려는 항목이 반드시 입력되어야 하는 지를 알 수가 없으므로 트랜잭션 처리기에서는 해당하는 템플리트 파일을 메모리에 상주시키면서 사용자에게 정보를 제공하게 된다.

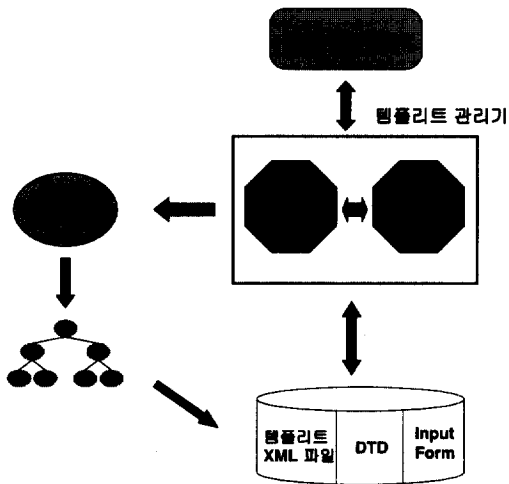
2.2.2 사용자 입력 폼

사용자 입력 폼은 사용자 데이터를 입력 받기 위한 컨트롤이다. 이 컨트롤은 XML파일형식으로 저장되며 이 파일을 읽어서 템플리트 관리기가 템플리트 파일과 태그를 비교해가면서 화면에 생성시킨다. DOM을 이용하여 파일의 관리가 용이하게 하기 위하여 XML파일

형식으로 저장하며 기존의 웹을 기반으로하는 프로그램에서는 이런 사용자 데이터를 받기 위해 웹 서버를 통하여 받을 수 있었으나 본 논문에서는 웹 서버를 사용하지 않고 일반 컨트롤을 사용하여 사용자 입력값을 처리하였다. 또한 이 사용자 입력 폼은 새로운 DTD가 들어오면 새로운 XML 템플릿 파일을 생성하면서 템플릿 관리기에 의해서 사용자가 직접 화면구성을 할 수 있도록 하였다.

2.3 템플릿 관리기

템플릿 관리기는 DTD를 입력 받아 템플릿 XML 파일을 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.



(그림 2.3) 템플릿 관리기

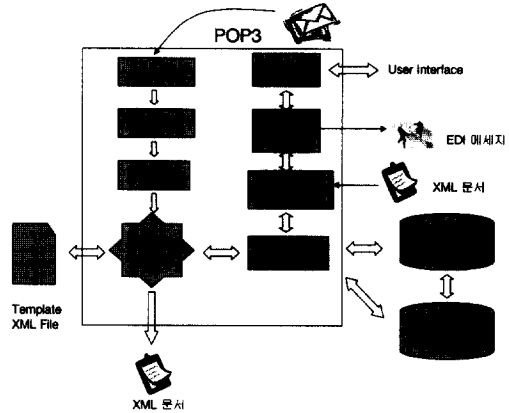
2.4 변환기

변환기는 기존 EDI와 XML 문서와의 호환을 위해 제공하는 기능으로 기존 XML로 문서를 변환하기 위해서는 EDI메시지를 이루는 세그먼트의 구조를 알고 있어야 한다. EDI 세그먼트와 엘리먼트를 XML문서로 표현하여 저장하여 세그먼트 구조를 제공하였다.

이러한 정보 엘리먼트 코드는 세그먼트로 정보를 제공하게 되며 변환과정에서 있을 수 있는 예외 처리를

위하여 정보를 제공한다.

본 시스템에서 제공하는 변환 기능은 XML문서를 EDI 메시지로 변환하는 것과 EDI메시지를 XML문서로 변환하는 것이다. 아래 (그림 2.4)는 변환기의 세부 구성도 이다.



(그림 2.4) 변환기 시스템 세부 구성도

본 논문에서 변환기는 XML문서를 EDI문서로 변환하기 위하여 XML문서를 속성추출기에서 속성과 사용자 입력값을 추출하고 구조분석기에서 기존 EDI문서 구조를 참조하여 구문규칙 적용기에 의해 EDI문서가 생성된다. 또한 EDI문서 생성시 발생할 수 있는 예외 처리를 위하여 예외처리기 존재한다. EDI 문서를 XML로 변환하기 위해서는 먼저 전송되어온 EDI문서를 수신기에서 수신하여 POP3통신규약의 머리부분을 제거하여 메시지의 정렬을 위해 정렬기로 보내어지고 EDI파서를 통해 XML 템플릿 문서를 참조하여 XML생성기에서 XML문서가 생성된다.

2.4.1 송수신기

전송하려는 문서를 SMTP(Simple Mail Transfer Protocol)통신규약으로 코딩하여 해당하는 메일 서버로 송신하며 수신시는 POP3 통신규약을 사용하여 수신 문서를 수신하게 된다.

수신하는 문서에 프로토콜의 헤더를 삽입하여 SMTP을 이용하여 자신의 메일 서버로 전송하고 수신된 문서는 POP3프로토콜을 이용하여 수신하고 헤더 부분을 제거한 후 필요에 따라 변환기로 보내 변환 후 XML 문서로 화면에 브라우저된다.

2.4.2 정렬기

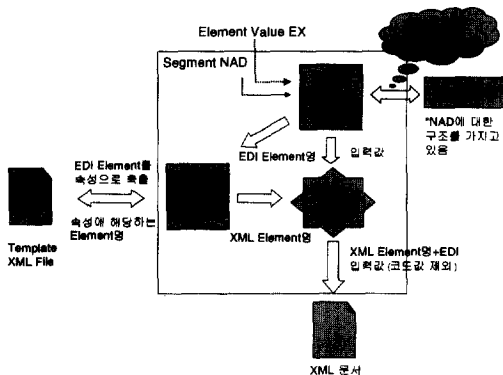
EDI메시지는 정렬되지 않고 연속된 문자열 형태로 되어있다. 이렇게 수신되는 메시지를 각 세그먼트별로 관독이 용이하도록 정렬한다. EDI메시지는 하나의 세그먼트로 시작하여 어깨점(‘)으로 끝난다. 세그먼트 단위로 정렬하기 위해서는 이러한 어깨점을 기준으로 정렬하면 된다. 그러나 EDI메시지의 내용 중에 어깨점을 사용하는 내용이 나올 수 있다. 이런 경우 잘못된 정렬을 하게 된다. 이러한 문제를 해결하기 위해서는 어깨점을 찾은 후 어깨점 전에 가능해제문자(?)가 있는지 확인하여 정렬해야 한다. 이렇게 정렬된 메시지를 EDI파서로 각 세그먼트 단위로 전송한다.

2.4.3 EDI파서

정렬기에서 세그먼트 단위로 정렬된 EDI메시지를 EDI파서는 엘리먼트 단위로 토큰화(Tokenizing)하여 XML생성기로 전송한다. EDI파서는 처음 “+” 문자를 만나기 전까지의 문자를 세그먼트로 보고 어깨점을 종료문자로 인식한다. 이러한 문자를 받은 XML생성기는 구조분석기에 해당하는 세그먼트 구조 정보를 요청하게 된다.

2.4.4 XML 문서 생성기

구조분석기의 정보와 템플릿 XML파일의 정보를 이용하여 XML문서를 생성한다.



(그림 2.5) XML 문서 생성기

● EDI 엘리먼트 추출기

EDI파서에서 받은 세그먼트명을 구조분석기로 보내고 해당하는 입력값은 XML생성기로 보냄

● 구조분석기

엘리먼트 추출기에서 받은 세그먼트명을 이용하여

XML형태로 저장된 세그먼트의 구조를 가져옴.

● XML 엘리먼트 추출기

EDI엘리먼트 추출기에서 받은 세그먼트명을 속성값으로 갖는 XML 엘리먼트를 템플릿 파일에서 가져옴.

● XML 문서 생성기

EDI 엘리먼트 추출기에서 받은 입력값과 XML 엘리먼트 추출기에서 받은 엘리먼트명을 이용하여 XML문서 생성

2.4.5 구조분석기

XML화되어 만들어진 EDI 세그먼트와 엘리먼트를 DOM을 이용하여 메모리에 적재하여 구조를 분석한다. (그림 2.6)은 XML화 되어있는 세그먼트와 엘리먼트이다.

```

Segment

<?xml version="1.0" encoding="EUC-KR"?>
<!-- edited with XML Spy v2.5 NT - http://www.xmlspy.com -->
<!DOCTYPE Segment SYSTEM "Segment.dtd">
<Segment EdiCode="NAD">
  <Name>상호 및 주소</Name>
  <Description>상호 및 주소는 계약대상자의 이름과 상호 주소</Description>
  <Element EdiCode="3035" ref="3035.xml" req="M">
    <Name>거래당사자 한 줄어</Name>
    <Description>거래당사자의 유형을 나타내는 코드</Description>
  </Element>
  <CompositeElement EdiCode="C082">
    <Name>거래당사자식별번호</Name>
    <Description>코드로 거래당사자를 식별하는 코드</Description>
    <Element EdiCode="3039" ref="3039.xml" req="M">
      <Name>거래당사자식별번호</Name>
      <Description>거래에 관련된 거래당사자를 식별하는 코드</Description>
    </Element>
    <Element EdiCode="1131" ref="1131.xml">
      <Name>코드목록한 줄어</Name>
      <Description>코드 목록의 식별</Description>
    </Element>
    <Element EdiCode="3055" ref="3055.xml">
      <Name>코드목록관리기관</Name>
      <Description>코드목록을 관리하는 기관의 식별코드</Description>
    </Element>
  </CompositeElement>
  ..
</Segment>
  
```

```

Element

<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE Element SYSTEM "Element.dtd">
<Element EdiCode="0007">
  <Name>당사자 식별 부호 한 줄어(Partner Identification code qualifier)</Name>
  <Description>당사자 식별 부호의 사용 기준을 나타내는 한 줄어</Description>
  <CodeList>
    <Value>1</Value>
    <Description>DUNS(Dun and Bradstreet)</Description>
    <Value>4</Value>
    <Description>ATA(International Air Transport Association)</Description>
    <Value>5</Value>
    <Description>INSEE/SIRET</Description>
    <Value>6</Value>
    <Description>Uniform Code Council Communications ID</Description>
    <Value>9</Value>
    <Description>DUNS with 4 digit suffix</Description>
    <Value>12</Value>
    <Description>Telephone Number</Description>
    <Value>14</Value>
    <Description>EAN (European Article Numbering Association)</Description>
    <Value>18</Value>
    <Description>NASE/SIREN</Description>
    <Value>20</Value>
    <Description>NADA(Automotive Industry Action Group)</Description>
    <Value>22</Value>
    <Description>ISO 8533 Organization Identification</Description>
    <Value>31</Value>
    <Description>DIN (Deutsches Institut fuer Normung)</Description>
    <Value>33</Value>
    <Description>BIA (Bundesversicherungsanstalt fuer arbeitslose)</Description>
    <Value>34</Value>
    <Description>National Statistical Agency</Description>
    ..
  </CodeList>
</Element>
  
```

(그림 2.6) 세그먼트와 엘리먼트

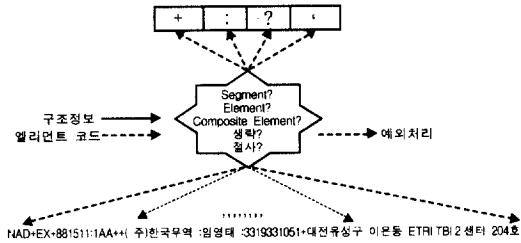
2.4.6 속성 추출기

XML문서를 EDI문서로 변환 시에 변환기로 입력된 XML문서를 DOM API를 이용하여 메모리에 트리형태로 표현한다. 구조분석기에서 요구하는 속성을 갖는 XML 엘리먼트 값을 구조분석기에 보내게 되며 구조분석기는 입력된 엘리먼트 값이 EDI세그먼트의 어떤 엘리먼트 값인지를 분석하여 구문규칙 적용기에 정보를 보낸다. 속성추출이 되지 않았을 때는 NULL값을 구문분석기로 보내 구문규칙적용기에서 해당하는 EDI 엘리먼트가 생략되었다는 정보를 제공하게 된다.

2.4.7 구문규칙 적용기

구조 분석기에서 제공하는 구조정보와 EDI 엘리먼트 코드값을 받아 EDI구문규칙을 적용한다. 구조 분석기에서 분석하는 세그먼트의 복합엘리먼트(Composite Element), 엘리먼트(Data Element)를 각각 인식할 수 있도록 구조 정보를 제공하면 축약의 규칙에 따라 EDI문서를 생성한다.

(그림 2.7)은 구문규칙 적용기를 나타낸 그림이다.



(그림 2.7) 구문규칙 적용기

2.4.8 예외 처리기

EDI 문서의 엘리먼트 중에는 사용자의 입력값이 아닌 순수하게 코드값으로 이루어진 엘리먼트가 있다. 이러한 엘리먼트는 XML문서 작성 시에 누락되거나 표현되지 못할 경우가 있다. 이런 경우 사용자 인터페이스를 통해 사용자가 직접 엘리먼트 코드의 내용을 보면서 코드값을 입력할 수 있도록 예외처리기를 설계하였다.

3. EDI문서 교환 시스템 구현

3.1 기반 환경

본 논문에서 설계 및 구현한 EDI문서 교환 시스템은 Windows NT 4.0 시스템을 사용하고 있으며 사용

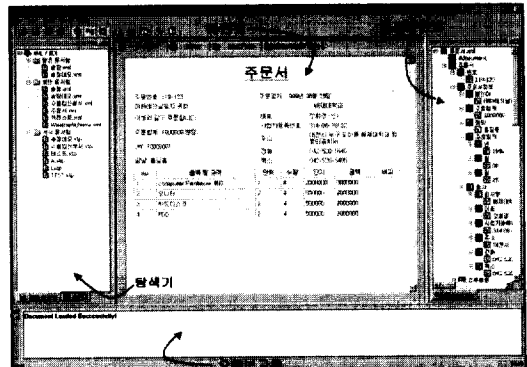
언어는 Visual C++ 6.0이다. MSXML 파서를 사용하여 XML문서를 검증을 하며 문서의 브라우징을 위하여 MFC(Microsoft Foundation Class)에서 제공하는 IE5.0 (Internet Explorer)컨트롤을 이용하여 브라우징을 처리하였다. 또한, 문서 생성 및 템플릿 생성과 관리를 구현 하기위해 DOM 인터페이스를 이용하였으며 문서의 송수신을 위하여 SMTP와 POP3를 이용하여 구현하였다.

3.2 사용자 인터페이스

사용자 인터페이스는 사용자가 편리하게 문서를 만들 수 있도록 사용자가 자신의 작업 창을 구성할 수 있도록 구현하였다. 크게 4가지 작업창으로 구성되었다.

첫 번째로, 왼쪽 상단의 탐색기 창은 사용자가 수신한 문서와 송신한 문서를 볼 수 있도록 트리 컨트롤로 처리하였으며 생성한 서식문서를 사용자가 이용할 수 있도록 하였으며, 오른쪽 상단의 엘리먼트 뷰는 XML 문서의 구조를 보여준다. 이 창은 서식 문서를 생성하기 위한 인터페이스로도 제공된다. 그리고 오류 보고 창은 파서에서 나오는 오류를 보여주는 곳으로써 수신 문서나 생성된 문서를 화면에 브라우징 할 때 파서에서 생성되는 오류메시지를 받아 이 창에 보여준다.

마지막으로 화면 중앙의 메인 뷰는 세 가지 탭 컨트롤로 되어 있으며 첫 번째 탭의 Viewer는 XML문서를 브라우징 할 수 있으며 HTML문서도 가능하며, 두 번째 탭의 Editor는 브라우징시에 오류가 발생한 문서를 편집할 수 있는 창이다. 이 탭에서 간단히 수정할 수 있는 것은 이곳에서 수정한다. 세 번째 탭의 Input Form은 사용자가 서식 문서를 선택하였을 때 해당하는 입력폼



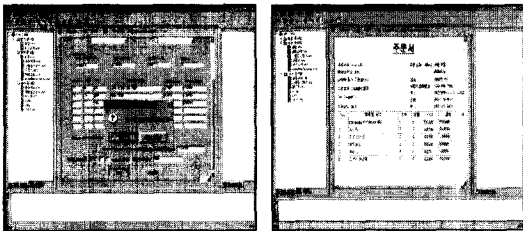
(그림 3.1) 사용자 인터페이스 구현

을 메모리에 적재하여 화면에 표시하는 창이다. 이곳에서 사용자가 컨트롤에 데이터 값을 입력하여 새로운 문서를 생성할 수 있다.

네 번째 탭은 Input Form이 없는 새로운 DTD가 입력되면 새로운 Input Form을 생성하지 않고 트리컨트롤과 리스트 컨트롤을 이용하여 데이터를 입력할 수 있도록 하였다. (그림 3.1)은 시스템의 사용자 인터페이스이다.

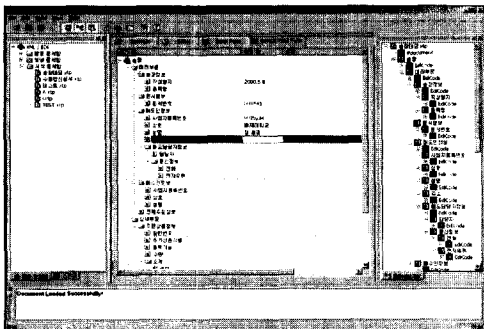
3.3 트랜잭션 처리기

트랜잭션 처리기는 사용자에게 입력값을 받아 XML 문서를 생성하는 것을 담당한다. 사용자가 원하는 템플릿과 파일을 선택하면 입력폼이 화면에 표현되고 표현된 입력폼에 데이터를 삽입하여 문서를 생성하고 전송한다. (그림 3.2)는 사용자 입력폼에 데이터를 삽입 후 전송하는 그림과 전송 후 XSL을 이용하여 화면에 브라우징 한 것이다.



(그림 3.2) 트랜잭션 처리기 구현 - 데이터를 삽입 후 전송(좌), 화면에 미리보기(우)

(그림 3.3)은 입력폼이 없는 새로운 DTD에 대하여 서식문서를 생성한 후 사용자에게 데이터를 입력 받는 그림이다. 기존에 생성된 입력폼 이외에 새로 생성된 서식문서에 대하여 입력폼을 따로 만들 필요없이 입력



(그림 3.3) 트랜잭션 처리기 구현 - 입력폼 값 입력

가능하도록 구현하였다.

3.3.1 트랜잭션 처리기 주요 Class와 함수

본 Class와 함수는 전송할 문서를 생성하고 처리하는 과정에서 사용하는 주요한 Class와 함수에 관한 것이다.

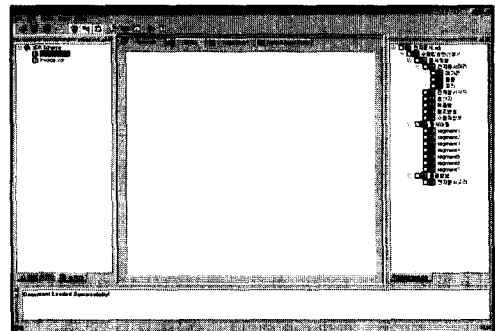
<표 3.1> 트랜잭션 처리기 주요 Class 명과 함수 명

| 클래스명/함수명 | 기능 |
|----------------------------------|--|
| CDomMange/ Trans() | 경로(path)명의 타입인 BSTR타입을 VARIANT구조체 형식으로 변환하여 파일명 리턴 |
| CCreateDocument/ Info() | DTD정보를 입력폼에 제공 |
| CCreateDocument/ Insertnode() | 입력값을 템플릿 노드에 입력 |
| CCreateDocument/ WriteDoc() | 메모리에 있는 문서를 파일로 생성 |

Trans()라는 함수는 DOM 인터페이스를 사용할 때마다 VARIANT 타입으로 변환하여 사용하는 것을 하나의 함수로 구현하여 중복되는 선언을 제거하였다. Info()라는 함수는 템플릿 문서정보를 입력 받아 사용자 입력폼에 정보를 제공한다. 사용자 입력폼에서 받은 데이터를 메모리상에 있는 노드에 삽입하는 함수이며 WriteDoc()은 이것을 파일로 생성하는 함수이다.

3.4 템플릿 관리기

템플릿 관리기는 사용자가 전송하기 원하는 엘리먼트를 선택하여 DTD정보를 포함하는 템플릿 파일을 만들 수 있으며 이 템플릿 파일에 기반하여 입력폼을 생성 및 수정할 수 있도록 구현하였다.



(그림 3.4) 템플릿 관리기의 구현

위 그림은 사용자가 전송하기 원하는 엘리먼트들을

선택하는 그림이다. 이렇게 선택된 엘리먼트는 템플릿 파일로 생성하게 된다.

이렇게 생성된 템플릿 파일을 편집기 창에서 열어 볼 수 있으며 선택된 엘리먼트들에 대해 템플릿 파일이 생성된다.

3.4.1 템플릿관리기 주요 Class 및 함수

본 Class와 함수는 템플릿 파일을 생성하는 과정에서 사용하는 주요한 Class와 함수에 관한 것이다.

〈표 3.2〉 템플릿 관리기 주요 클래스 및 함수 명

| 클래스명/함수명 | 기능 |
|----------------------------------|---|
| CElementBar/ InsertDTDnode() | 사용자에게 DTD정보를 읽어서 화면에 트리형태로 만들어 주기 위한 함수 |
| CElementBar/ ViewElement() | 트리형태로 DTD를 보여주는 함수 |
| CCreateDocument/ Info() | 사용자에게 DTD정보를 알려주기 위한 사용자 인터페이스 |
| CCreateDocument/ Insertnode() | 체크된 엘리먼트를 node에 추가 |
| CCreateInputForm | 사용자 입력폼의 생성을 위한 클래스 |
| CDomManage/ Load() | 템플릿 문서를 적재하는 함수 |

처음 DTD의 트리를 만들기 위해서 InsertDTDnode()를 호출하여 DTD를 트리형태로 만들게 되며 ViewElement()함수를 이용하여 화면에 보여주게 된다. Info()함수를 이용하여 사용자가 엘리먼트를 선택할 때 DTD의 정보(필수, 조건)를 사용자에게 알려주며 체크된 엘리먼트를 Insertnode()함수를 이용하여 메모리에 추가한 후 파일로 생성하게 된다.

4. 고 찰

4.1 고 찰

본 논문에서는 XML에 기반하여 EDI문서를 효율적이고 유동적으로 교환하기 위한 시스템을 설계하고 구현하였다. DTD를 입력 받아 템플릿을 생성할 수 있도록 하여 사용자의 입력을 받을 수 있도록 하였고 이런 서식에 대하여 직접적으로 사용자에게 일반 컨트롤을 사용하여 표현하였다.

사용자에 의해 입력 받은 데이터를 XML문서로 생성시키도록 하기위해 생성되는 XML문서의 트랜잭션을 트랜잭션 처리기를 두어 처리하도록 하였다. 이러한 처리를 위해서는 문서의 데이터 값을 접근하여 처리할 수 있어야 하기 때문에 W3C에서 표준으로 정의

된 DOM을 사용하였다. DOM은 HTML과 XML문서를 위한 API로써 문서를 접근하고 관리하기 위한 방법을 API를 통해서 제공한다[9].

KEDIFACT 표준전자문서를 이용하여 DTD를 개발하였으므로 기존 EDI메시지에 가깝게 XML문서를 만들 수 있도록 하였으며 문서의 송수신을 위하여 SMTP와 POP3 통신규약을 사용하였다.

또한, 기존 EDI와 호환되도록 하기 위해서 XML문서와 EDI문서 상호간에 변환을 할 수 있는 변환기를 설계하였다. XML문서의 효율적인 브라우징을 위하여 IE 컨트롤을 사용하였으며 그 구조를 볼 수 있도록 트리구조로 구현하였다. 또한, 사용자 취향에 따라 자신의 작업환경을 구축할 수 있도록 각각 윈도우를 도킹이 가능하도록 구현하여 사용자 편의를 최대한 고려하였다.

4.2 본시스템의 장점

본 시스템의 장점은 첫째 웹에 기반하지 않고 독립 응용프로그램으로 설계 및 구현 되었기 때문에 HTTP 통신규약상의 보안 문제를 제거할 수 있으며 둘째 웹에 기반하여 고정적인 서비스를 제공 받는 것이 아닌 사용자가 직접DTD에 따라 문서를 생성 및 관리 할 수 있도록 한 것이 특징이다.

셋째 아직까지 타 시스템에서 문서의 변환을 위한 해법을 구하지 못한 상태에서 본 시스템은 XML을 이용하여 EDI문서의 구조정보를 제공함으로써 해법을 제공하였다[16, 20].

기존 국내 웹 기반으로 개발된 시스템의 경우 DTD를 자동생성할 수 있도록 제공하고 있다. 하지만 이 시스템은 DTD 자체가 XML의 장점을 살리지 못하고 EDI표준 메시지 태그를 그대로 수용하여 생성하고 있다. 이것은 XML 엘리먼트가 자기 자신을 기술하는 장점을 사용하지 못하여 엘리먼트 명 만으로 어떤 엘리먼트인지 알 수 가 없다.

또한, 국내 타 시스템이 기존 시스템과의 호환에 관하여 고려하고 있지 않고 있는 반면 본 시스템에서는 변환에 반드시 필요한 EDI메시지 구조를 XML형식으로 저장하고 이것을 DOM API로 관리하도록 하여 XML의 구조정보를 최대한 이용함으로써 기존 시스템과의 호환을 고려하였다[16].

본 논문에서 보완해야 할 부분은 XML문서를 암호화 할 수 있는 암호화기가 필요하며 문서의 논리적 흐름에 따라 사용자를 통하지 않고 문서를 자동생성 할 수 있는 에이전트기술이 보완해야 할 점이다.

4.3 다른 시스템과의 비교

기존에 개발되어진 시스템은 아직 XML을 이용한 EDI문서 처리과정이 표준화 되어 있지 않기 때문에 많지 않다. 기존의 웹기반 시스템으로는 국내에서 개발된 시스템이 있으며 VAN을 사용하지 않는 EDI로는 Internet EDI가 있다. 아래 <표 4.1>은 위 두 가지 시스템과 본 시스템을 비교한 것이다[16, 20, 21].

<표 4.1> 타 시스템과의 비교

| | 국내 타 시스템 | Internet EDI | 본 시스템 |
|------------|----------------------|-----------------------|---------------------------------|
| Protocol | HTTP | HTTP | SMTP, POP3 |
| XML사용 여부 | 사 용 | 비사용 | 사 용 |
| 보안성 | HTTP의 특징상 보안문제 해결과제 | HTTP의 특징상 보안문제 해결과제 | 메시지 자체의 암호화만 필요함 |
| 문서표현 | CSS를 사용 | CSS를 사용 | XSL을 사용 |
| 구조적 검색 여부 | 어려움(Tag를 EDI태그를 사용함) | 불가(HTML은 구조적 검색이 불가함) | 가능(XML의 구조적 특징을 이용함) |
| 자유로운 문서 생성 | 제공된 HTML입력폼에 따라 | 입력, 제공된 HTML입력폼에 따라 | 사용자의 엘리먼트 선택에 따르는 자유로운 문서 생성 가능 |
| 확장성 | 가 능 | 불 가 | 가 능 |

주 : < Internet EDI의 경우는 관련된 시스템이 여러개 있으며 모두 공통된 형태를 취하고 있어서 통합한 개념으로 비교하였으며 국내의 경우는 Internet EDI로 구성하지 않은 제품이 있어 특화해서 비교하였음.>

자료 : 참고 [16, 20, 21]

위의 <표 4.1>에서 보듯이 본 시스템의 경우는 기존의 EDI 시스템들이 일반적으로 취하는 형태 중에서 장점에 해당하는 부분들을 섭렵하고 단점에 해당하는 부분들을 직접 구현하거나 다른 해법을 제시함으로써 기존 EDI 시스템들이 가지는 단점을 극복하였다.

4.4 결론 및 발전방향

본 시스템의 경우는 XML/EDI의 강점을 최대한 살리고 있으며 개발된 표준수립, 자체적인 트랜잭션 처리, 유동적인 사용자 입력을 위한 인터페이스 제공, 구현이 쉽고, 신뢰적이고, 메시지 자체에 대한 보안 모듈을 사용할 경우 상당히 견고한 보안 정책 수립도 수월하다.

본 연구 결과로 인해 그 동안 EDI를 사용해 왔던 기업은 XML을 사용하여 EDI를 확장할 수 있게 되며, 그 동안 경제적인 이유 때문에 EDI를 이용할 수 없었던 중소기업들은 편리하게 EDI를 이용할 수 있게 될

것이다. 또한 본 시스템을 확장하여 전자결제 시스템으로도 이용할 수 있으며 이에 따라 능률적인 기업환경을 만들 수 있으리라 사료된다[7, 8].

향후 연구 과제는 첫째 현재까지 XML/EDI가 표준화 되어 있지 않기 때문에 시스템의 각 처리기를 모듈화 하여 다른 시스템에서 사용가능 하도록 해야 하며 둘째로는 문서가 기업의 비즈니스 문서이기 때문에 보안을 유지하기 위한 시스템이 필요함으로 차후에 문서를 암호화 하여 보낼 수 있는 연구가 필요하다. 셋째는 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.

또한, XML 문서의 태그가 있는데 이 태그 문서의 증가에 의한 트랜잭션의 크기가 증가한다. 이것은 XML/EDI가 처리를 위해 더 많은 대역폭을 쓴다는 것이다. 실제로 문자수에서 전통적인 EDI의 트랜잭션보다 50%이상이 더 높다[3, 4, 15]. 그러나, 현재의 통신 대역폭은 H/W의 발달로 인해 큰 제약점이 되고 있지 않으며, 태그 문자의 증가는 얻을 수 있는 많은 이점에 비하면 매우 작은 것이 된다.

참 고 문 헌

- [1] 정희경, "XML 가이드", 도서출판 그린, 1999.
- [2] 채규혁 번역, 차세대 웹의 혁명 XML, 도서출판 대림, 1998.
- [3] 林承宅, "EDI 革命으로 가는 길", 대청출판사, 1997.
- [4] 문희철, 심상렬 공저, "무역자동화와 EDI", 貿易經營社, 1998.
- [5] 한국전자거래표준원, "한국 EDI 표준코드집", 한국전자거래표준원, 1998.
- [6] Charles F. Goldfarb, Paul Prescod, "The XML Handbook," Prentice-Hall, 1998.
- [7] Frank Boumphrey 외 12명, "XML Applications," Wrox Press, 1998
- [8] Alex Homer, "XML in IE5 Programmer's Reference," Wrox Press, 1990.
- [9] <http://www.w3.org/DOM/> "The DOM Level 1 Specification".
- [10] http://www.keb.or.kr/sub_3/sub3_1.htm "EDI의 개요".
- [11] http://www.keb.or.kr/sub_3/sub3_3_3.htm "한국공업규격 전자문서 구분 규칙 (KS X 6706)".
- [12] http://www.keb.or.kr/sub_3/sub3_3_4.htm "표준 디렉토리".

- [13] http://www.keb.or.kr/sub_3/sub3_3_1.htm "표준 전자문서".
- [14] <http://www.geocities.com/WallStreet/Floor/5815/guide.htm>.
- [15] <http://www.xmledi.net> "Guidelines for using XML for Electronic Data Interchange".
- [16] http://www.dib.net/4th_dib/kor/2_1.htm#4 "DIB XML/EDI Solution".
- [17] <http://www.commercedesk.com/content/standards/default.asp?tab=Edifact>"Un/EDIFACT Overview".
- [18] <http://www.disa.org/>.
- [19] http://www.commerceone.com/solutions/xml/xml_tech.htm "XML and CBL".
- [20] <http://www.xml.com/pub/1999/08/cdi/index.html> "Introduction to XML and EDI".
- [21] <http://time.kotel.co.kr/s-trends/200006/인터넷EDI표준-양성현.htm> "인터넷EDI표준".



임 영 태

e-mail : youngpc@nobreak.com

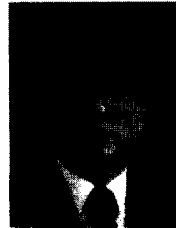
1997년 홍익대학교 전자전산공학과 학사

1999년 홍익대학교 전자전산공학과 석사(네트워크 전공)

1997년~1999년 삼성소프트웨어 멤버십

1999년~현재 배재대학교 컴퓨터공학과 박사과정 재학중
노브레이크 테크놀러지스(주) 재직중

관심분야 : 실시간 처리 시스템, 분산 미들웨어 시스템,
인터넷상의 멀티미디어, 게임



한 우 용

e-mail : wyhan@etri.re.kr

1983년 경희대학교 공과대학
전자공학과 학사

1985년 경희대학교 대학원 전자
공학과 석사

1989~현재 한국전자통신연구원,
컴·소프트웨어 연구소, 전
자상거래 연구부, 전자지불
연구팀

관심분야 : 분산 객체 컴퓨팅 기술, XML 전자상거래
공용 프레임워크, XML 기반 B2B 시스템
통합, XML 디지털 서명



정 회 경

e-mail : hkjung@mail.paichai.ac.kr

1985년 광운대학교 컴퓨터공학과
(학사)

1987년 광운대학교 컴퓨터공학과
(석사)

1993년 광운대학교 컴퓨터공학과
(박사)

1994년~현재 배재대학교 컴퓨터공학과 교수

관심분야 : 하이퍼미디어/멀티미디어 문서정보처리, SGML,
XML, HyTime, DSSSL, IETM XML/EDI,
ebXML