

XML 기반의 e-비즈니스 문서 생성을 위한 폼 생성시스템

김 성 한[†] · 김 창 수^{††} · 정 회 경^{†††}

요 약

본 논문에서는 e-비즈니스 DTD(Document Type Definition) 문서를 기반으로 한 XML 폼 생성기(Form Generator)를 설계 및 구현하였다. 인터넷 보급과 정보환경 변화는 e-비즈니스에 있어서 많은 변화를 제공하였고, 거래에 사용될 비즈니스 문서 양식에 있어서도 웹 형식이거나 전자화된 전용 양식 형태로 변환될 필요가 있게 되었다. 이러한 상황에서 각각의 업체별로 다양한 문서 포맷을 사용하여, 상호주체 간에 포맷 호환성 제공을 위한 필요성이 제기되며, 따라서 재사용성과 포맷 호환성 지원에 따른 비용의 문제를 안고 있다. 이에 따라, 논문에서 제안하는 XML 폼 생성기는 XSLT를 이용하여, 생성 결과인 XML 폼 문서에 HTML(HyperText Markup Language) 형태의 웹 문서로 생성 후에, 사용자 입력을 통해 e-비즈니스 DTD 문서 구조에 유효한 XML 비즈니스 메시지를 작성할 수 있다.

XML Based Form Document Generation System for e-Business

Sung-Han Kim[†] · Chang-soo Kim^{††} · Hoe-kyung Jung^{†††}

ABSTRACT

In this paper, XML form generator is designed and implemented on the basis of e-business's DTD (Document Type Definition) document. Rapid evolving for internet services and information infrastructure give many impacts on the e-business, it need to make a new kinds of web-based or electronic-based document formats for e-business transaction trading. In current situations, there are many kinds of document formats on conventional business documents for each companies. And, it has many problems on the aspects of the document reusability and cost to support interoperability between documents for the trading partners. To solve this interoperability of documents, the constructed XML form generator is changing XML form document into HTML (HyperText Markup Language) based web document by XSLT. And it also generates XML business message validating for e-Business DTD by user inputs.

키워드 : e-비즈니스(e-business), XML 폼 생성기(XML form generator), 비즈니스 거래(business transaction)

1. 서 론

산업의 주축이 정보 중심으로 변화되어가고 있는 시점에서, 비즈니스 거래(Business Transaction)에 대한 처리가 전자화 되고, 자동화되어, e-비즈니스라는 보다 확대된 개념으로 정의되고 있다. 또한, e-비즈니스 상의 전자 비즈니스 문서의 활용은 비약적으로 증가되고 있으며, 거래 주체의 독립적인 사용에 머물지 않고 거래 주체들간의 교환을 통한 문서의 활용에 이르기까지 그 영역이 확대되고 있다. 이러한 e-비즈니스 환경 하에서 전자화된 비즈니스 문서 사용을 위해서는 문서 제작을 위한 문서 양식(Form)이 필요하지만 전통적인 종이 형태의 문서 양식은 제작 시간과 처리 절차의 고비용 문제점을 가지고 있으며, 생성한 문서 양식의 활용에 대한 면에 대해서도 기존 시스템과 거래 주

체들과의 상호운용성의 처리 문제를 지니고 있다. 전자의 문제점은 문서 양식의 생성에서 활용, 보관, 재활용에 이르는 과정에 있어서 여러 단계의 과정을 거쳐야 하고, 보관 역시 종이 형태의 파일 저장이 이루어져 관리의 어려움이 존재하게 되며, 유사한 문서를 작성시에도 기존의 문서를 활용하기 어려워 재 디자인을 하여야 하는 문제를 갖고 있다. 후자의 문제에 있어서도, 수작업으로 작성된 종이 형태의 비즈니스 문서는 현재의 상황과 같이 자동화되고 일원화된 비즈니스 처리에서 효율을 발휘하기 어려운 문제를 안고 있다.

본 논문에서는 XML(eXtensible Markup Language)을 이용하여 인터넷 비즈니스에 사용될 문서 양식을 생성하고 XML 문서의 포맷팅을 위해 제안된 인터넷 문서 표준인 XSL(eXtensible Stylesheet Language)을 이용하여 XML 문서를 변환하여 보편화된 웹 문서 표준인 HTML(HyperText Markup Language) 문서로 생성한다. 이 문서를 통해, 거래 주체간에 이용되어질 XML 비즈니스 메시지를 생

† 정 회 원 : 한국전자통신연구원 표준연구센터 근무
 †† 준 회 원 : 배재대학교 IT 센터 책임강사
 ††† 종신회원 : 배재대학교 정보통신공학부 교수
 논문접수 : 2002년 1월 21일, 심사완료 : 2002년 4월 25일

성하게 된다. 따라서, XML 기반으로 생성된 웹 문서는 기존의 종이 형태나 특정 포맷의 문서 양식이 갖는 생성과 활용에 대한 문제점을 해결 할 수 있게 될 것이며, XML 비즈니스 메시지를 통해 거래 주체간의 원활한 상호 운용성을 갖게 할 수 있을 것이다[1].

본 논문의 구성은 다음과 같다. 2장에서는 전자 상거래에서 활용되고 있는 기존 시스템에 대한 분석 및 제안 시스템의 특징을 소개하고, 3장에서는 시스템의 설계 내용을 설명하며, 4장에서는 구현과 고찰에 대한 기술 및 마지막으로 5장에서는 결론 및 향후 과제를 제시한다.

2. 관련 기술 연구

본 장에서는 전자 거래에서 활용되는 기존 시스템의 주요 특징 분석 및 제안한 시스템의 장단점을 기술한다. 기존의 고 비용을 투자해 개발한 전자상거래는 응용 및 엔티티간의 상호 운용성이 지원되어야 하는 문제가 있다. 따라서, XML이 갖는 데이터 내용 정의 기능과 정보교환 기능을 활용하여 기존 어플리케이션간의 비즈니스 정보 공유 문제들을 해결하는 기술을 제시되고 있다. 본 절에서는, 현재 기존의 EDI 기술 및 XML 기반의 프레임워크의 종류에 대해 기술하며, 본 시스템과의 장단점을 제시하였다.

2.1 기존 상용 시스템의 분석

2.1.1 EDI(Electronic Data Interchange)

EDI는 정형화된 거래, 행정 관련 정보를 데이터 기술상의 일정한 통신표준에 입각하여 컴퓨터와 컴퓨터간에 구조화된 전자 표준문서나 파일 형태로 교환, 축적, 처리하는 방식이다. 즉 업무 문서를 기계적으로 처리할 수 있는 구조화된 형식으로서 컴퓨터 통신에 의해 조직 상호간에 교환하는 방식이 EDI이다. 서류 문서를 전자파일 형태로 생성하여, 종래의 우편, 전화, 인편에 의한 전달 방식 대신 전자적인 방식으로 전송하는 것이라고 규정할 수 있다. EDI는 UN/EDIFACT (United Nations Rules for Electronic Data Interchange for Administration, Commerce and Transport)란 행정, 무역 및 운송을 위한 전자문서교환에 관한 국제연합규칙이다. 이것은 유럽경제 위원회(UN/ECE)에 의해 UNTDID(UN Trade Data Interchange Directory)로 승인, 공포 되었으며, 합의된 절차에 따라 유지, 개선되고 있다.

EDI 기술의 확산을 저해하는 장애 요인으로는 구축비용, 법적인 문제, 메시지의 불안정성, 시스템 구축의 복잡성, 거래처리 약정 체결 과정에 소요되는 비용과 시간, 비용 대비 효과의 미흡, 보안성, 거래 상대방마다 고유한 시스템 구축, 향후 발전 방향의 불확실성 등을 들 수 있다. 또한 기업 환경은 급격하게 변화하고 있고 이에 따라 EDI에 요구되는 경영상의 요구 사항들이 추가적으로 발생하고 있는 것을

따라 가는 것이 어려운 점이다[2].

2.1.2 전자상거래 프레임워크의 개요

전자상거래 프레임워크란 컴퓨터 및 인터넷망을 이용하여 한 기업체의 비즈니스 경계를 넘어 거래 상대자(Trading Partner)와의 상거래를 가능하게 하는 기능을 규정하는 일반적인 골격을 의미하며 BizTalk, eCo, RosettaNet 및 ebXML 등이 있다. 각 프레임워크의 목적은 다음과 같다.

BizTalk 프레임워크의 목적은 XML의 도입을 더욱 가속화하는 데 있다. 여기에는 XML 스키마의 구현을 위한 디자인 프레임워크와 어플리케이션들 간에 메시지를 주고받기 위한 XML 태그들의 집합이 포함된다. 마이크로소프트를 비롯한 여러 소프트웨어 회사들과 표준 단체들은 BizTalk 프레임워크를 일정한 양식의 XML 스키마를 생성하는데 이용하고 있다.

eCo 프레임워크는 인터넷 전자상거래 시스템 개발 관련 회사들 중심으로 1994년 CommerceNet이라는 연합 컨소시엄을 구성 하여 인터넷 전자상거래 공용 프레임워크 표준을 만드는 프로젝트이다. eCo 프레임워크의 목적은 필요한 정보를 찾고 기업간 정보기술 방법들을 제시하고 업계 및 사용자들이 기본 정보를 연장해서 자신의 비즈니스에 부합하게 하는 것이다. eCo 규격은 인터넷에서 서로 다른 전자상거래 시스템간의 상호 운용을 촉진하기 위해 만들어졌다.

RosettaNet의 목적은 폭 넓은 산업과 개방적인 전자상거래 표준을 만들고 구현하기 위해 구성된 비영리적인 기관으로 모든 공급체인거래 상대자들과 구매자들의 이익을 제공하기 위해 공급체인거래 상대자들간에 전자적인 비즈니스 인터페이스를 제공하여 개방적이며 일반적인 프로세스를 정의하기 위한 것이다. RosettaNet 구현 프레임워크 규격에서 주요 부분은 개방적이고 유연한 전자상거래 프로세스의 실행을 정의하고 선도하기 위하여 OBI(Open Buying on the Internet)기술 규격에 따라 정의되고 RosettaNet의 주요 프로세스들은 파트너사이의 e-비즈니스 인터페이스를 배열하여, 궁극적으로 구매자들과 공급체인의 파트너들과 상호 작용할 수 있고 조율할 수 있게 하는 것이다[3].

ebXML 목적은 전세계의 정보교환을 가능하게 하는 XML 기반의 개방형 인프라 제공을 목표로 안전하고 보편적이며, 정보처리의 상호 운용이 가능한 표준 기술 규격을 연구, 개발, 보급하여 단일한 전세계 전자상거래 시장을 구축하는 것이며, 현재 표준화 진행 중이다[4].

2.2 제안 시스템의 특징

본 시스템은 EDI 문서를 기반으로 한 기존시스템과 비교해 데이터의 정형화로 인한 확장성의 문제점을 XML을 기반으로 한 문서 편집시스템을 개발하여 해결하였으며, 문서 변환 시스템을 활용하여 기업간 데이터 교환을 쉽게 함으로써 기업간 비즈니스 정보공유를 원활하게 하였다. 또한,

기업간 정보공유를 위해 사용되는 비즈니스 XML 메시지 생성에 있어서 웹 폼을 통해 생성함으로써 거래 주체간에 편리성과 상호 운용성을 제공할 수 있다.

실세계에서 기존의 전자상거래 프레임워크들이 상호 운용성을 강조하면서도 이를 지원하지 못하고 있는 실정이지만 본 논문에서 제안한 시스템은 기업간 비즈니스 정보공유를 위해 자동화되고 단일화된 전자상거래 환경을 구축하여 전자상거래 환경에서의 상호운용성의 문제점을 해결하고 있다.

3. 폼 생성 시스템 설계

본 논문에서 제안하는 전체시스템은 인터넷 문서 표준인 XML을 이용하여 e-비즈니스 상에서 사용될 웹 폼 문서를 생성하기 위한 것으로, WYSIWYG(What You See Is What You Get)한 사용자 인터페이스를 통하여 XML 폼 문서를 생성하기 위한 폼 생성기(Form Generator)와 작성된 XML 폼 문서를 변환시키기 위한 변환기(Form To HTML Translator), 웹 문서 형태의 폼 문서를 통한 XML 비즈니스 메시지를 생성하는 처리기(Processor Of Creating Biz XML Message)로 구성된다. (그림 1)은 시스템의 전체 구성도이다.



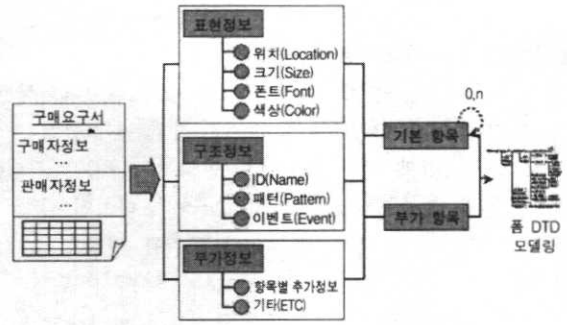
(그림 1) 시스템 전체 구성도

3.1 XML 폼 생성기 설계

입력된 e-비즈니스 문서 DTD를 통해 표현하고자 하는 문서 폼 요소를 사용자의 의도에 따라 디자인하고 수정하는 과정을 거친 후, 생성시킨 폼 문서는 객체화를 위해 XML 형태 폼 문서로 저장한다. 폼 생성기는 폼 모델링을 위한 DTD 설계 부분과 e-비즈니스 DTD를 통한 객체 소스 트리(Object Source Tree) 생성 설계, 그리고 폼 디자인을 위한 편집 모듈 설계 부분으로 구성된다[5].

3.1.1 XML 폼 DTD 설계

비즈니스 문서의 폼 작성은 표현적인 면에서 차이점을 보이고 있지만 폼 내부에 구성된 콘텐츠는 일관성을 유지하는 경우가 많다. 따라서, e-비즈니스 상에서 사용되어질 XML 폼 문서를 위한 DTD 설계에도 이점을 고려하여 설계했다.

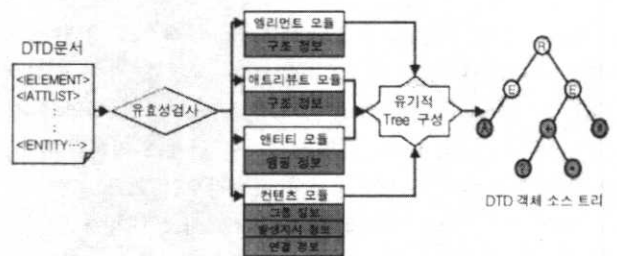


(그림 2) XML 폼 DTD 모델링

(그림 2)와 같이 임의의 문서 폼으로부터 폼을 구성하는 각각의 항목을 추출한다. 각 항목에는 위치, 크기, 폰트, 색상 등의 표현정보가 존재하며, 각 항목의 구분자(Identifier)가 될 ID(Name)와 항목의 계층적 정보를 갖는 패턴, 동적 요소인 이벤트 등의 구조 정보를 갖는다. 또한, 항목들을 구별하는 부가 정보를 포함한다. 이러한 세 가지 분류 정보를 기반으로 하여, 폼 구성의 필수 정보를 구성하고 반복 여부를 분석하여 폼 구성의 기본 항목으로 정의한다. 기본 항목을 추가적으로 지원하기 위해 부가 항목을 정의한다. 기본 항목과 부가 항목을 재구성하는 과정을 통하여 페이지 단위의 폼 DTD가 모델링 된다[6, 7].

3.1.2 DTD 객체 소스 트리 설계

XML 기반의 폼을 생성하기 위해서는 입력 데이터인 e-비즈니스 DTD를 디자인 모듈에서 사용할 수 있도록 트리 형태의 객체 소스로 재구성하고 기본 문서 처리를 위한 해당 DTD 문서의 유효성 검증을 한다.



(그림 3) DTD 객체 소스 트리 생성

DTD에는 해당 문서를 정의하기 위해 엘리먼트, 에트리뷰트, 엔티티의 단위 정보들과 그룹정보, 발생시자 정보, 연결자 정보 등으로 구성된다. 이러한 정보를 (그림 3)에서와 같이 각각 모듈 단위의 정보 객체로 추출하게 되고, 추출한 정보의 유기적인 재구성을 통해 트리 형태의 DTD 객체 소스 트리를 생성시킨다.

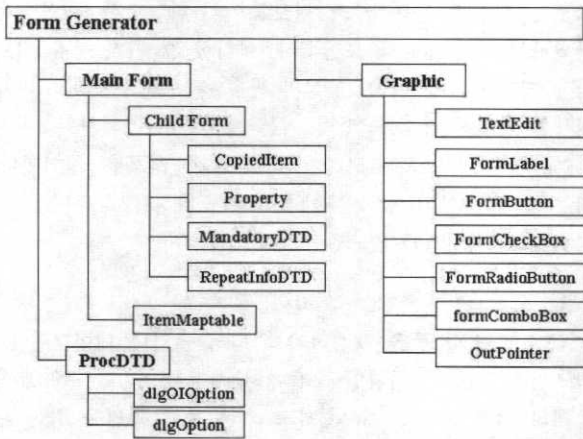
3.1.3 XML 폼 생성 모듈 설계

XML 폼 DTD를 통해, 디자인 될 폼 문서 구조를 정의하고, e-비즈니스 문서 DTD의 정보를 트리로 갖고 있는

DTD 객체 소스 트리로 디자인 할 폼 항목들의 계층 정보와 구분 정보를 제공하게 된다.

폼의 디자인은 사용자 인터페이스를 통해 제공되어진 디자인 영역에 WYSIWYG한 방식으로 작성된다. 설계 주관점에서도 제시했던 것과 같이 디자인되어 질 폼은 인터넷상의 웹 폼을 기준으로 하고 있다. 따라서, 디자인 할 폼의 항목들도 HTML 웹 폼의 항목들과 동기를 이루도록 설계하였다[8].

(그림 4)는 폼 생성기의 클래스 구성도를 나타내고 있다. 'Child Form' 클래스는 디자인 될 영역을 제공하게 되며, 디자인을 위한 행위 즉, 마우스와 키보드 등을 통해 이루어지는 행위에 대한 전체적인 처리를 정의하고 있다. 관련 클래스의 객체화 역시 'Child Form' 클래스를 통해 수행되어진다.



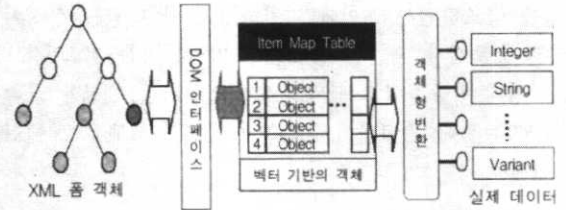
(그림 4) 폼생성기 클래스 구성도

'CopiedItem' 클래스는 'Child Form' 클래스의 영역에 그려진 선택된 디자인 항목을 복사 및 삭제를 위한 클래스 정의가 이루어진다. 'Property' 클래스는 디자인 영역의 선택된 디자인 항목에 대한 수정을 위한 클래스 정의이다. 이 클래스를 통해서 디자인 항목의 표현정보인 크기, 폰트, 색상 등의 정보 변경을 하게 되며, 구조적이고 내용정보인 구분자, 패턴, 이벤트 항목에 대해 변경과 정의를 수행한다.

'ProcDTD' 클래스는 이전 절에 기술한 DTD 객체 소스 트리 생성을 위한 클래스이다. 또한, 이 클래스를 통해서 비즈니스 XML 메시지 생성을 위한 비즈니스 문서 DTD의 템플릿 구성 과정을 정의하게 된다. 디자인 영역에 실질적인 그래픽 객체를 제공하기 위한 클래스 정의로써, 'TextEdit', 'FormLabel', 'FormButton', 'FormCheckBox', 'FormRadioButton', 'FormComboBox', 'OutPointer'와 같은 클래스 정의가 요구되어진다. 각 클래스들은 웹 기반의 폼 항목을 주축으로 하고 있으며 표현 요소인 크기, 폰트, 색상, 텍스트 등의 정보를 통해서 디자인 영역에 그래픽적인 형상을 구현할 수 있도록 정의한다. 'OutPointer' 클래스는 디자

인 항목의 복사, 갱신, 삭제 등의 수정 사항에 대해 각 항목의 선택과 항목 구분자를 추출하도록 정의한다.

(그림 5)는 폼 생성기 내부의 자료 구조를 나타낸다.



(그림 5) 폼 생성기 내부 자료구조

자료 구조를 담당하는 클래스는 (그림 5)에 제시된 'Item-MapTable' 클래스이다. 폼 생성기 내부의 데이터는 그래픽 표현을 위한 데이터 형이 실수, 정수, 문자열, Variant 형과 같이 여러 형태로 나타나게 되고 데이터의 동적인 변동 역시 매우 크게 나타난다. 이러한 점을 대처하기 위해, 'Item-MapTable' 클래스는 (그림 5)에 나타난 것과 같이 벡터 기반의 객체 구성을 하게 된다. 벡터 자료 구조는 데이터 형을 'Object' 형으로 구성한다. 따라서, 다양한 데이터 형을 일반화 할 수 있는 장점을 가지게 된다. 뿐만 아니라, 벡터는 배열과 달리, 고정된 데이터 크기를 가질 수 있는 동시에, 무정의 데이터 크기에도 능동적인 대처가 가능하다.

디자인된 폼 항목들은 XML DOM(Document Object Model) 인터페이스를 통해서 XML 노드 객체로 구성되며, 해당 DOM 노드는 트리 구조로 되어 있게 된다. 이 트리를 통해서 외부 출력과 입력이 이루어진다.

3.1.4 폼 문서 유효성 검증 및 반복 구문 처리 설계

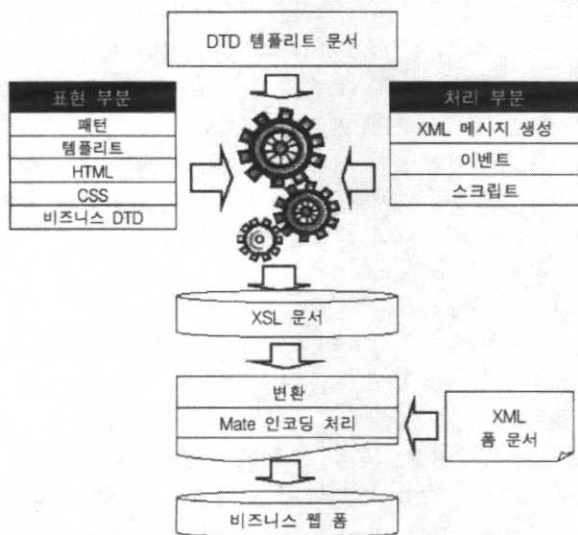
e-비즈니스 문서 DTD를 기반으로 한 폼을 생성하는 것을 기본으로 하고 있으므로 디자인된 폼의 항목이 해당 DTD의 필수 요소 정보를 포함하고 있는지 검증하는 절차가 필요하게 된다. 해당 DTD로부터 객체 소스 트리를 구성하는 과정에서 검증에 요구되는 데이터 즉, 계층적 정보와 그 구분자를 추출하고, (그림 4)의 'MandatoryDTD' 클래스로 그 정보를 보관하게 된다. XML 폼 문서를 저장하는 시점에서 이 검증 절차를 처리하게 된다. 'Mandatory-DTD' 클래스 객체의 데이터와 XML 형태로 재구성한 폼 항목에 대해 매핑 과정의 비교를 통해 유효성 검증을 하게 된다. 이 과정에 의해, 작성된 폼 문서는 비즈니스 DTD에 요구되는 필수 항목을 반드시 포함하는 역할을 수행하게 된다.

비즈니스 문서를 포함하는 폼에는 해당 항목에 따라 반복적인 구문이 요구되는 경우가 많다. 이러한 반복구문 처리를 위해, (그림 5)의 'RepeatInfoDTD' 클래스를 정의하여 폼 구성에 존재하게 될 반복 항목들의 구조적 정보와 구분자를 각각 추출하여 보관하게 되고 반복 이벤트를 처리하

는 항목에 대한 속성 정보를 갱신하는 과정에서 'Repeat-InfoDTD' 클래스의 객체로부터 반복 항목을 추출하여 웹 형태의 스크립트 함수의 매개변수로 제공하게 된다.

3.2 XML 폼 문서 변환기 설계

생성된 XML 폼 문서를 활용하기 위해서 HTML 형태의 웹 폼으로의 변환 처리가 이루어져야 한다. 변환 대상은 폼 생성기를 통해 생성된 XML 폼 문서이고, 이 문서를 포맷팅하고 구조 변환시키기 위해 XSL 문서를 이용하게 된다. (그림 6)은 XML 폼 문서 변환기의 구성 흐름도를 나타내고 있다.



(그림 6) XML 폼 문서 변환기 구성 흐름도

3.2.1 XSL 문서 설계

(그림 6)의 상위 부분에서 보여지는 것과 같이, XSL 문서의 설계는 XML 폼 DTD에 의거하여 작성되게 된다. 기본 단위로 분류한 폼 상의 항목들을 공통 요소와 개별 요소로 추출하여 XSLT의 변환 기본 단위인 템플릿 규칙으로 구성하게 된다. 각 단위의 템플릿 규칙은 해당 패턴에 따라, 포맷팅 요소가 될 HTML, CSS의 구문 적용이 이루어지고, 필요한 문서 내용을 추출하기 위한 XPath의 기본 함수를 포함하게 된다. 이러한 처리는 폼 생성기를 통해 생성되는 모든 XML 폼 문서에 적용된 일반적 사항이다. 폼 항목들의 개별적인 사항은 이벤트 정의를 통해서 수행하게 된다. 이벤트 항목도 웹 폼 문서에서 정의되는 이벤트와 동일하게 구성한다. 이벤트의 정의에 따른 웹 스크립트 처리 내용을 XSL 문서에 포함시켜 XML 폼 문서를 변환시킬 XSL 문서를 작성한다[9, 10].

3.2.2 폼 문서 변환 처리

폼 생성기를 통해 얻어진 XML 폼 문서를 앞 절에서 제시한 XSL 문서를 통해 e-비즈니스 웹 폼으로 변환하는 과

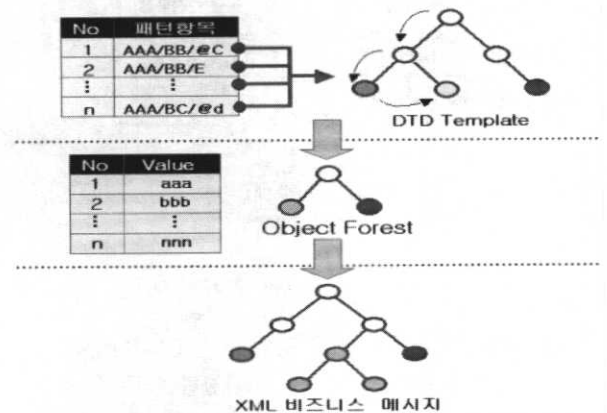
정이다. (그림 6)의 하부의 처리에서처럼 변환 과정을 거치게 되는데 XML 문서 조작을 위한 인터페이스인 DOM 인터페이스를 통해 변환과정을 거치게 된다. 변환과정은 크게 두 부분으로 나누어져 변환이 이루어진다. 소스 문서 즉, XML 폼 문서의 각 항목 패턴은 변환 문서의 템플릿 규칙에 따라 변환되어 웹 문서의 폼 항목들로 재구성되고, 변환 문서 내부의 이벤트 스크립트들은 'CDATA' 형태의 데이터로 처리되어 목적 문서인 웹 폼에 포함되게 된다. 변환이 완료된 후, 문서의 인코딩 문제에 독립적 대응을 하기 위한 처리가 수반되어야 한다.

3.3 비즈니스 XML 메시지 생성 처리기 설계

e-비즈니스 웹 폼 문서로부터 e-비즈니스 상에 교환될 XML 메시지 생성은 본 논문에서의 가장 궁극적인 목적이라 할 수 있다. 인터넷 상의 보편적인 문서 형태인 HTML을 이용 한다는 점은 폼을 이용하는 사용자들이 인터넷 HTML 폼에 익숙하여 XML에 대한 부담을 주지 않을 것이며, 사용에 편리성을 제공하게 될 것이고, 사용자 입력의 결과가 비즈니스 상에서 사용 할 XML 메시지라는 점은 B2B 환경의 비즈니스 업무 처리에서 고용량의 데이터 처리와 상호 운용성, 기존 시스템과의 연동 등에서 유용한 처리를 가능하게 해 줄 것이다.

3.3.1 XML 비즈니스 메시지 생성 모듈

(그림 7)에서와 같이, XML 형태의 비즈니스 메시지의 생성은 폼 변환과정을 거쳐 생성된 HTML 문서의 폼 항목들에 대하여 구분할 수 있는 이름과 패턴 정보, 생성하고자 하는 XML 문서의 구조를 갖는 DTD 템플릿 문서를 이용하게 된다. HTML의 폼 객체인 'Document' 객체를 이용하여 폼 항목의 이름과 패턴 정보를 배열 형태로 구성한다. 구성된 배열의 각 정보는 생성하고자 하는 비즈니스 문서의 구조를 갖고 있는 DTD 템플릿을 통해 검증 및 매핑 과정을 거치게 된다. 검증과정은 패턴 형태의 구조 정보를



(그림 7) XML 비즈니스 메시지 생성 모듈

DTD 템플릿의 구조에 깊이 우선 방식(DFS : Depth First Search)으로 탐색하여 구조의 진위 여부를 가리게 된다. 이 과정을 거친 후, 매핑이 이루어진 구조는 하나의 객체 트리(Object Forest)로 재구성되고 폼에 입력된 실제 데이터와 결합하게 된다. 이러한, 객체 트리들 간의 재 구조화를 통해 e-비즈니스 DTD에서 요구되는 구조를 갖는 XML 메시지를 작성한다.

문서 내 반복 구조에 대한 처리는 HTML 문서 내 반복되어질 패턴 정보를 수집하여 해당 스크립트 함수의 매개 변수에 전달한다. 매개변수의 내용으로 객체 트리를 생성하고 해당 객체 트리가 유효한 구조로 존재하는지 확인한다. 존재하는 구조일 경우, 실제 데이터가 입력 상황에 맞추어 순차적으로 XML 객체를 생성 추가한다.

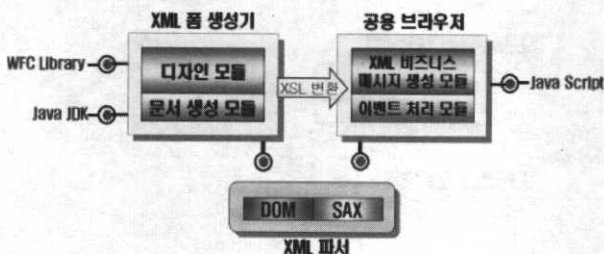
3.3.2 폼 문서 상의 이벤트 처리 모듈

폼 생성 과정에서, 폼의 각 항목에 대한 속성을 정의하는 과정을 통하여 해당 폼 항목이 HTML 폼 문서로 변환되어졌을 때 발생될 여러 이벤트를 정의한다. 각 이벤트 발생에 따른 처리 모듈인 스크립트는 XSL을 이용하여 변환시키는 과정에서 HTML 문서에 포함된다.

4. 구현 및 고찰

4.1 구현

본 논문에서 제안한 시스템은 IBM-PC 호환 컴퓨터(Pentium III-800)에서 개발하였으며, Windows 98과 Windows 2000의 운영체제 환경에서 Java JDK(Java Development Kit) 1.1을 탑재한 Microsoft visual J++ 6.0(Developer Studio 98 Enterprise Edition)을 사용하여 구현하였다. XML과 XSL 문서의 검증과 조작을 위해 DLL(Dynamic Link Library) 형태로 제공되는 Microsoft사의 MSXML 3.0 파서를 이용하였다. (그림 8)은 본 논문에서 구현된 XML 폼 생성기 시스템의 전체 시스템 구성도를 나타내고 있다.



(그림 8) 시스템 구성도

(그림 8)에서와 같이 구현 범위는 폼을 디자인하고, 그 결과인 XML 폼 문서를 웹 문서로 변환하는 독립적인 어플리케이션 영역과 변환된 웹 폼 문서로부터 비즈니스에 사용될 XML 메시지를 생성하기 위한 공용 브라우저 영역으

로 구분된다.

XML 폼 생성기는 Microsoft 사의 WFC(Windows Foundation Class)를 통해 윈도우 구성요소를 지원 받게 되며, Java JDK를 통해 내부 알고리즘을 구현하였다. 공용 브라우저를 통한 XML 비즈니스 메시지 생성과 이벤트 처리를 위해 자바 스크립트를 이용하였다. 또한, XML 객체의 처리를 위해 MSXML 파서의 개발 인터페이스인 DOM과 SAX (Simple API for XML)를 이용하였다[11].

4.1.1 DTD 객체 소스 트리 구현

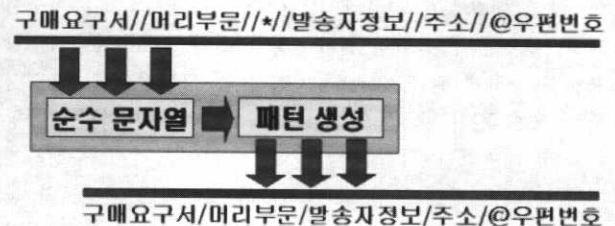
생성하고자 하는 폼의 내용적, 구조적 정보를 가지고 있는 e-비즈니스 DTD 문서를 입력받아 폼 디자인을 위한 자료로 이용될 객체 소스 트리로 구현하였다. DTD 문서를 토큰으로 구분하여 'ELEMENT', 'ATTRIBUTE', 'ENTITY', 'CDATA', '#PCDATA' 등의 구분 객체로 분리하고, 구분된 객체를 최소 단위의 트리 객체로 구성한 후, 전체적으로 최소 단위의 트리 객체를 유기적으로 재 구조화하였다. 재 구조화 과정에서 발생하는 트리 구조의 재귀적인 생성은 사용자 인터페이스를 두어 제어하였다.

4.1.2 폼 생성기 시스템 구현

DTD 객체 소스 트리로부터 생성 시킬 폼 문서의 유효성과 반복 사항 처리를 위해 각각의 정보를 깊이 우선 방식의 탐색법으로 추출한다.

폼 문서의 유효성 검증을 위해서는 필수적으로 제공되어야 하는 정보가 존재하며, 마지막 단말 노드의 상위 계층정보 상에*(repeat)나 '(option)'등의 구분자가 존재하지 않는 단말 노드를 추출하여 벡터 형태의 객체로 구성한다. 반복 사항 처리를 위한 반복 정보 추출 과정도 필수 정보 추출과정과 유사하며, '+'(plus)와 '*' 구분자를 갖는 경로를 찾아 그 단말 노드를 추출한다.

디자인 영역에 폼 항목들을 디자인하는 과정은 크게 선택된 폼 항목의 초기화 설정과 초기화된 폼 항목들의 수정 및 기능성 부여, 문서의 저장 과정으로 구분된다.

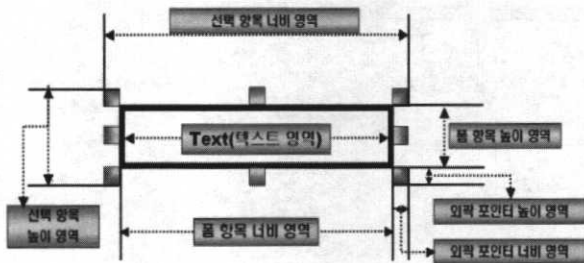


(그림 9) 패턴 추출 과정

폼 항목의 초기화는 우선, DTD 객체 소스 트리로부터 작성하고자 하는 폼 항목의 이름과 패턴을 추출한다. (그림 9)는 패턴을 추출하는 과정을 나타내고 있다. (그림 9) 상단의 문자열은 객체 소스 트리에서 추출된 경로 정보이다. 이

정보를 XPath의 패턴 형태로 변환하기 위하여 경로 정보를 단일 문자 단위로 분리 후, '+', '*', '?', '/' 등의 문자를 제거한 순수 문자열로 재 구성한다. 반복 과정을 통해서 얻은 순수 문자열을 경로에 맞게 구성함으로써 (그림 9)의 하단 형태의 패턴으로 생성되게 된다.

'Textedit', 'Label', 'Button', 'ComboBox' 등의 폼 항목들은 사용자에 의해 선택되어져 폼 항목의 이름과 패턴을 가지고 디자인 영역에 해당 폼 항목을 그리게 된다. 기본적으로 디자인 영역에 폼을 디자인하는 인터페이스는 마우스를 사용하였고, '마우스 다운', '마우스 이동', '마우스 업' 이 세 단계를 이용하여 디자인하게 된다.



(그림 10) 표현 영역

(그림 10)은 디자인 영역에 그려진 폼 항목의 표현 영역을 나타내고 있다. 표현 영역은 폼 항목 영역, 선택 폼 항목 영역, 외곽포인터 영역, 텍스트 영역으로 구성된다. 폼 항목이 선택되어질 때는 폼 항목 영역을 통해서 구분되어진다. 이러한 과정을 통해 폼 항목이 초기화된다.

폼 항목의 초기화가 설정된 후, 해당 폼 항목의 수정이나 기능성을 부여하게 된다. 수정 사항은 폼 항목의 위치, 크기의 변경을 할 수 있으며, 색상이나 폰트 정보를 변경함으로써 표현을 다양하게 구성 할 수 있다. 각 폼 항목은 이벤트 발생이 가능하기 때문에 해당 폼 항목의 이벤트 종류와 이벤트 발생에 따른 처리 내용을 정의함으로써 폼 항목의 기능성을 제공하고 있다.

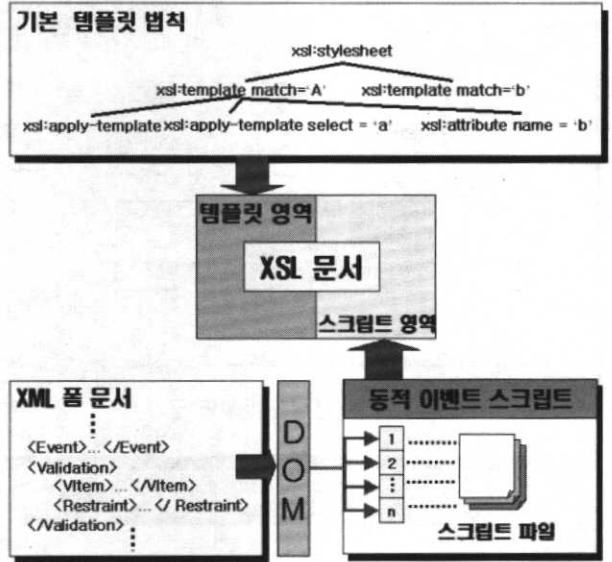
디자인이 완료된 폼은 저장된다. 저장하기 전에 폼 유효성 검증 과정을 거치며, 저장은 DOM 인터페이스를 이용하여 'ItemMapTable' 클래스 객체의 벡터화 된 폼 항목 정보를 XML 문서 형태로 생성한다.

4.1.3 웹 폼 문서 변환기 구현

폼 생성기로부터 생성된 XML 폼 문서를 정형화된 변환 템플릿을 가진 XSL 문서로 변환하여 웹 폼 문서로 생성하는 과정이다. 변환기의 처리 과정은 변환을 위한 XSL 문서의 구성 및 생성 모듈과 생성된 XML 폼 문서를 XSL 문서에 의한 변환하는 모듈로 구성된다.

(그림 11)은 변환을 위한 XSL 구성 및 생성 과정은 기본 템플릿 규칙과 동적 이벤트 처리 스크립트로 구성된다. 정형화된 정보인 기본 템플릿 규칙들은 생성된 모든 XSL

문서에 포함된다. 기본 템플릿 규칙은 변환될 폼 항목의 외형적인 변환 속성과 폼 항목의 구분자 정보를 웹 폼에게 전달하는 변환 속성으로 구성되어 있다.



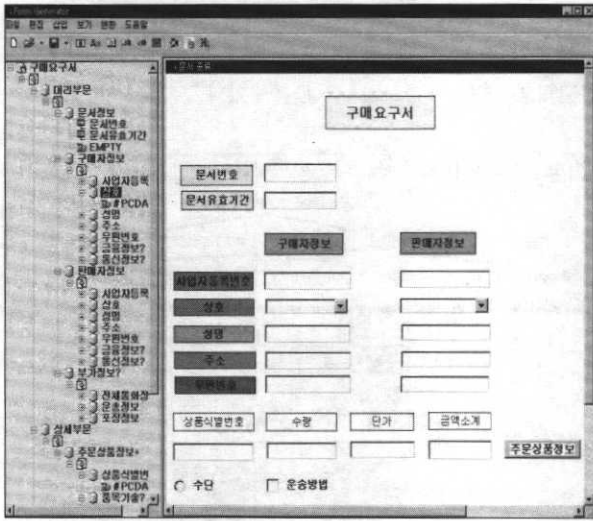
(그림 11) XSL 구성 및 생성 과정

동적 이벤트 처리 스크립트는 파일 단위로 구성되어 있는 이벤트 스크립트들이 XML 폼 문서의 'VItem' 엘리먼트의 내용 구분자에 의해서 호출되어 CDATA 형태로 XSL 문서 안에 포함된다. 뿐만 아니라, 폼 문서 내의 반복 정보를 갖고 있는 'RepeatScriptInfo' 객체 정보는 처리 되어질 스크립트 함수의 매개변수 값으로 전달된다.

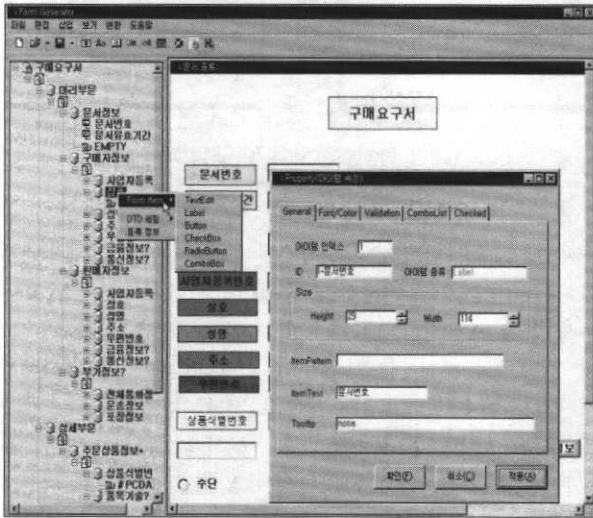
생성된 XML 폼 문서를 XSL 문서에 의해서 변환하는 모듈은 MSXML 파서가 지원하는 XML DOM 인터페이스의 확장 메소드인 'transformNode()'을 이용하여 변환하게 된다. 이 메소드는 XSL의 DOMDocument 객체 메소드으로써, DOMDocument 객체를 매개변수로 하며 반환하는 값은 변환된 내용의 문자열을 전달한다. 이 과정에서, 'transformNode()' 메소드는 메타 인코딩 기준을 'UTF-8' 포맷으로 변환되게 된다. 따라서, 한글 처리의 문제가 발생하게 된다. 이 문제를 해결하기 위해, 'modifyHTML()' 메소드를 개발하여 특정 메타 인코딩에 의존하지 않도록 구현하였다.

4.1.4 폼 생성 시스템 구현 결과

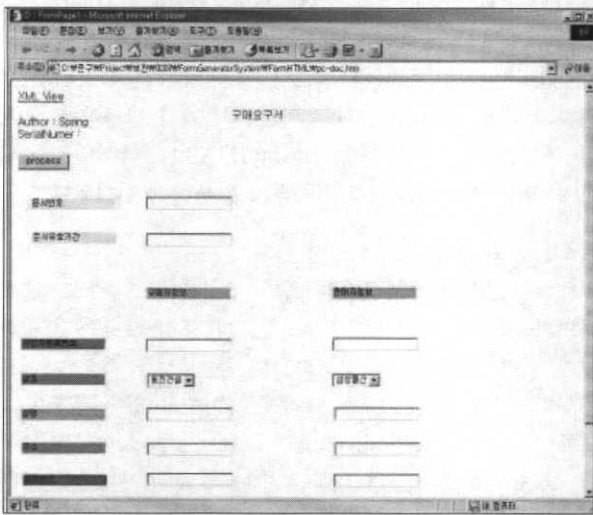
(그림 12)는 폼 생성기의 화면 구성을 보여 주고 있다. 화면의 왼쪽에 보여지는 트리 형태의 구성 요소는 생성하고자 하는 비즈니스 문서 DTD를 폼 생성을 위한 객체로 생성하는 DTD 객체 소스 트리를 보여주고 있는 부분이다. 오른쪽에 보여지는 구성 요소는 DTD 객체 소스 트리로부터 폼 항목을 사용자의 의도에 따라 폼을 디자인하는 영역으로써, WYSIWYG한 방식으로 폼을 작성 및 수정하면서 폼을 제작하도록 하였다.



(그림 12) 폼 생성기의 화면 구성



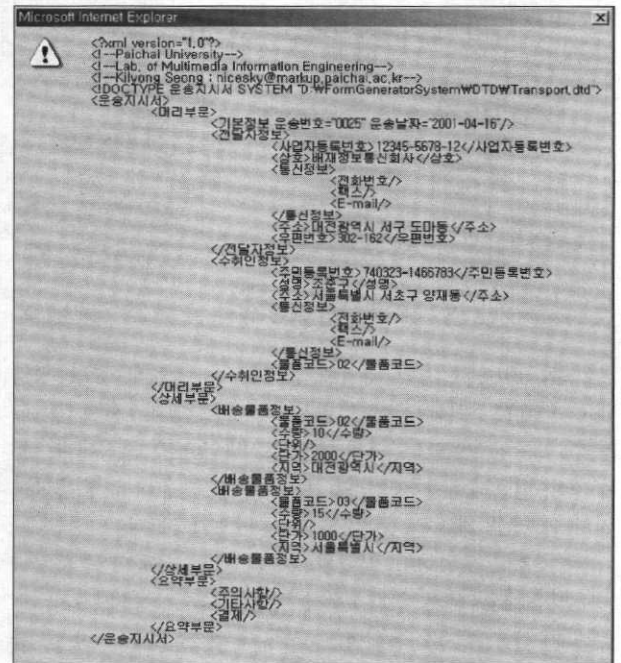
(그림 13) 폼 생성기의 속성 창과 선택 메뉴



(그림 14) 변환된 웹 폼 문서

(그림 13)은 폼 생성기의 속성 창과 DTD 객체 소스 트리의 선택 메뉴를 보여주고 있다. 폼 생성기로부터 생성된 XML 폼 문서를 기본 템플릿 법칙과 이벤트 스크립트를 포함한 XSL 문서를 통해 변환하여 (그림 14)에 보여지는 것과 같은 웹 폼 문서를 생성한다. 해당 웹 폼 문서는 비즈니스 DTD에서 요구하는 필수 비즈니스 항목을 포함하고 있으며, 각 항목에 대한 이벤트 스크립트를 포함하고 있고, 사용자 입력 결과를 XML 비즈니스 메시지로 재구성을 할 수 있도록 구현하였다[12, 13].

(그림 15)는 웹 형식의 폼 문서에 사용자 입력 후, 생성되어진 XML 메시지를 보여 주고 있다.



(그림 15) XML 비즈니스 메시지 생성 결과

4.2 고 찰

본 논문은 인터넷 문서 표준인 XML을 이용하여 e-비즈니스 상에 사용되어질 문서의 폼을 XML 형태로 생성시키는 폼 생성기와 생성된 XML 폼 문서를 XML 변환 언어인 XSL을 이용하여 HTML 형태의 웹 폼으로 변환하도록 하며, 이를 통해 비즈니스 XML 메시지를 생성하는 시스템의 설계 및 구현에 관한 것이다.

본 논문에서 제안한 폼 생성 시스템은 비즈니스 DTD 문서로부터 생성시킨 객체 소스 트리로부터 웹 폼 문서상의 구성된 폼 항목을 기반으로 폼을 디자인하게 되며, XML/XSL 문서의 검증과 조작을 지원하기 위한 MSXML 파서를 통해, 폼 정보를 저장, 변환 및 XML 비즈니스 메시지 생성에 이용하게 된다. 생성된 폼 문서가 XML 형태로 존재하도록 하여 폼 문서의 수정 요구에 능동적인 대응이 가능하도록 구현하였고, 변환을 통해 생성된 비즈니스 웹 폼

또한, XML 비즈니스 메시지 생성 스크립트와 각 이벤트 발생에 따른 처리 스크립트를 웹 폼 내부에 구성하여 웹 서버와 같은 각종 웹 프레임에 종속되지 않도록 구현하였다.

본 시스템은 비즈니스 DTD 문서를 기반으로 하여 사용자 주관에 의해 폼이 디자인되지 않고 DTD 객체 소스 트리를 통해 제어되고 규정된 폼 정보를 갖는 XML 문서를 생성할 수 있다. 또한, 웹 폼 문서의 결과가 e-비즈니스 프레임워크 상의 문서 교환 표준으로 주목받는 XML 메시지로 구성되어 비즈니스 거래 주체간의 문서 교환의 편리성과 상호 운용성을 제공하는 계기를 마련하였다. 앞으로 보안되어야 할 사항은 'Table', 'Image' 같은 다양한 폼 항목을 표현하기 위한 객체가 추가되어야 하고, 이원화된 XML 폼 문서 변환과정의 일원화하는 기술이 보완해야 할 점이다.

4.3 타 폼 생성 시스템간 비교

기존의 폼 제작 관련 소프트웨어에서 XML의 등장은 폼 관련 새로운 흐름을 제공하였다. 개별적인 폼 템플릿 포맷에서 벗어나 공용적이고 확장성을 가진 XML을 이용하게 되었다. XML을 기반으로 한 폼 개발 연구는 폼 제작을 위한 객체 소스의 제공 여부와 폼 저장 방식, 활용 측면으로 구분하게 된다. 일반적으로 폼 제작을 위한 객체 소스 형태는 폼 작성자의 개별적인 의견과 기존의 폼 양식의 참조를 통한 폼 제작이 이루어진다. 이러한 정보를 바탕으로 폼을 디자인하고 XML 형태로 저장하여 자체 표현 엔진을 통해 보여지게 된다. 기존의 폼 생성기와 본 시스템의 특징 및 장점, 단점을 비교해 <표 1>에 보였다.

<표 1> 타 폼 생성 시스템간 비교

시스템	특징 및 장점	단 점
타 시스템	<ul style="list-style-type: none"> 폼 항목에 대한 표현정보 및 연산기능(Computing function)을 포함하여 폼 자체 기능성이 높음 XFA 폼 표준을 통한 폼 디자인 데이터베이스, 웹연동을 통한 기능성 제공 	<ul style="list-style-type: none"> 비즈니스 문서 구조에 의거한 XML 메시지 생성 불가 공용 브라우저 사용을 위한 Plug-in 설치 필요
본 시스템	<ul style="list-style-type: none"> 비즈니스 DTD에 의거한 규정된 폼 생성 가능 비즈니스 프레임워크에 연동 가능한 XML 메시지 생성 가능 비즈니스 스크립트를 통한 융통성있는 업무 처리 가능 일반적인 웹 문서로 구성되어 시스템에 독립적인 구성 가능 	<ul style="list-style-type: none"> 테이블 형태의 다양한 폼 디자인을 지원하지 못함 문서 변환에 따른 부하 발생

5. 결 론

정보 통신 환경의 급변화는 기존의 비즈니스 체계를 빠른 속도로 탈바꿈시키고 있으며, 보다 효율적인 체계로의 개선을 요구하고 있는 상황으로 비즈니스에 사용될 문서 양식의

전자화, 효율화에 대한 관심이 고조되고 있는 상황이다.

이에, 본 논문에서는 인터넷 문서 표준이며, 문서 교환 언어의 대안으로 떠오르고 있는 XML을 이용하여 비즈니스에서 사용될 폼을 디자인하고 XML 형태로 저장하며, XML 폼 문서의 표현과 구조 변환을 위해 스타일 언어인 XSL을 이용하여 웹 환경에 사용할 수 있는 HTML 형태의 폼 문서로 변환하고, 변환된 폼 문서로부터 XML 비즈니스 메시지를 생성 할 수 있는 비즈니스 폼 생성 시스템을 개발하였다. 따라서, 기존의 종이 형태나 특정 포맷의 문서 양식이 갖는 여러 단계의 제작 과정과 그에 따른 비용의 문제, 서로 다른 문서 포맷으로 인한 거래 주체 간에 문서 교환을 통한 상호 운용성을 확보 할 수 없었던 문제들이 본 시스템의 개발을 통해서 해결될 수 있을 것이다.

본 논문에서 제안한 폼 생성 시스템은 e-비즈니스 DTD 구조에 의거한 정형화된 폼 문서를 생성할 수 있고, 일반 사용자에게 친숙한 HTML 문서 형태를 가지고 있어 특정 시스템에 종속되지 않는 특징과 비즈니스 시스템에서 범용적으로 사용하게 될 XML 비즈니스 메시지 작성에 대해 편리성을 제공하는 장점을 가지고 있다.

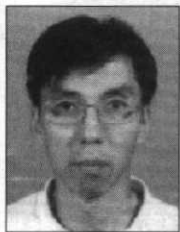
본 논문의 결과로 인터넷 e-비즈니스 시스템, 무선 인터넷 서비스 분야에 XML 기반의 전자 문서 생성을 위한 폼 개발에 유용하게 이용되리라 사료된다.

향후 연구되어야 할 부분으로는 폼 생성기를 통해 디자인 된 폼과 실제 HTML 형태로 변환된 폼과의 차이가 존재하고 있다. 이 차이를 테이블 형태의 동적 구성 요소의 적용을 통해 줄여나가야 할 것이며, 다양한 웹 요소의 추가가 요구되어진다. 뿐만 아니라, 이미 사용되어지고 있거나 개발 중인 비즈니스 시스템 프레임워크 생성된 폼 문서와의 연동을 원활하게 처리 할 수 있는 인터페이스에 대한 연구 개발이 필요하리라 사료된다.

참 고 문 헌

- [1] 한국 커머스넷, 이정열, "eCommerce 글로벌 리포트", 다우출판사, 2000.
- [2] UN/EDIFACT Overview, "http://www.commercedesk.com/content/standards/default.asptab = Edifact".
- [3] RosettaNet, http://www.rosettanet.org.
- [4] Alan kotok, David R. R. Webber, "ebXML," New Riders, 2001.
- [5] Aoki, Yoshinori, "Creating Interactive Web Forms from XML Document," XML2000.
- [6] W3C, Extensible Markup Language(XML) Version 1.0 (Second Edition), http://www.w3.org/TR/REC-xml, Oct., 2000.
- [7] 정희경, "WWW 문서 작성을 위한 차세대 언어 XML 가이드", 그린.

- [8] XFDL Designer, <http://www.pureedge.com/resources/xfdl.htm>.
- [9] W3C, XSL Transformations(XSLT) Version 1.0, <http://www.w3.org/TR/xslt>, Nov. 1999.
- [10] Neil Bradley, "The XSL companion," ADDISON-WESLEY, 2000.
- [11] W3C, Document Object Model Level 1, <http://www.w3.org/TR/REC-DOM-Level-1>.
- [12] 이형문, "XSL-fo를 적용한 XML 문서표현 시스템의 설계 및 구현", 배재대학교, 2000.
- [13] FormFlow, <http://www.jetform.com/e-Forms/formflow2.html>.



김 성 한

e-mail : sh-kim@etri.re.kr
 1985년~1989년 광운대학교 전자계산기공
 학과(공학사)
 1989년~1991년 광운대학교 전자계산기공
 학과(공학석사)
 1999년~현재 배재대학교 컴퓨터공학과
 박사과정

1991년~현재 한국전자통신연구원 표준연구센터 근무
 관심분야 : XML, 오디오 메타데이터 처리기술



김 창 수

e-mail : sungu@mie.paichai.ac.kr
 1996년 배재대학교 컴퓨터공학과 학사 졸업
 1996년~1998년 배재대학교 컴퓨터공학과
 석사 졸업
 1999년~2002년 배재대학교 컴퓨터공학과
 박사 졸업
 2002년~현재 배재대학교 IT 센터 책임강사
 관심분야 : 멀티미디어 문서정보처리, XML, XSLT, e-Business,
 Semantic Web



정 회 경

e-mail : hkjung@mail.paichai.ac.kr
 1981년~1985년 광운대학교 컴퓨터공학과
 (공학사)
 1985년~1987년 광운대학교 컴퓨터공학과
 (공학석사)
 1989년~1993년 광운대학교 컴퓨터공학과
 (공학박사)

1994년~현재 배재대학교 정보통신공학부 부교수
 관심분야 : XML, XSL, 멀티미디어 문서정보처리, 전자상거래,
 전자도서관