

---

# 전자상거래를 위한 XML 메시지 처리기 설계

## Design of XML Message Processor for Electronic Commerce

---

조광문  
천안대학교 정보통신학부

Kwang-Moon Cho(ckmoon@cheonan.ac.kr)

---

### 요약

XML이 W3C의 표준이고 플랫폼에 독립적이라는 특성 때문에 전자상거래에서 중요한 역할을 하고 있다. 이미 많은 전자상거래 시스템에서 XML을 이용하고 있다. B2B 전자상거래 분야에서는 기업간 비즈니스에 대한 업무 규칙과 절차가 표준화되어야 한다. 그러나 많은 기업들이 표준화된 XML 문서를 사용하는 것이 아니라 각 기업의 실정에 맞는 시스템을 구축하여 기업간 XML 문서 전달을 하고 있어, 이에 따라 문제점들이 나타나고 있다.

본 논문에서는 XML로 서비스를 요청하고 서버가 해당 서비스를 처리하는 XML 메시지 처리기를 설계하였다. 이 XML 메시지 처리기를 전자상거래 시스템에 적용하여 효율적으로 이용할 수 있을 것이다.

□ 중심어 : | XML 메시지 | 전자상거래 | XML 메시지 처리기 |

### Abstract

Because XML is a W3C standard and has characteristics like platform independent, it has a critical role in electronic commerce. Many electronic commerce systems are already using XML. Business rules and procedures should be standardized for efficient B2B electronic commerce. But a lot of companies have its own XML documents instead of standard documents.

In this paper an XML message processor is designed. Clients may request services in XML and server returns response in XML in this system. This XML message processor can be applied easily and used efficiently in electronic commerce systems.

□ Keyword : | XML Message | Electronic Commerce | XML Message Processor |

---

## 1. 서론

인터넷은 개방된 형태의 통신으로서 쉬운 프로토콜을 기반으로 한 브라우저와 다양한 정보 도구들이 제공되면서 인터넷을 통해 유통되는 정보가 많아지고 있다. 이에 따라 인터넷에서 사용되는 문서 처리 표준들

이 활발하게 개발되고 있으며 이에 기반한 전자상거래가 크게 확산되었다.

기업은 경쟁력 확보를 위하여 기업 내부의 조직 활동뿐만 아니라, 다른 기업과의 상호 협력적인 업무를 수행한다. 특히 기업간의 거래 업무는 계약된 문서 양

식을 주고받으며 협의된 업무 프로세스에 따라서 정형적으로 실행되는 것이 일반적이다. 기존의 기업 내부 프로세스를 관리하는 비즈니스 프로세스 관리 시스템을 바탕으로, 기업간 전자상거래에서 요구되는 여러 가지 핵심 요소들을 분석하여 B2B 환경에 적합한 환경과 모델을 구축하여야 한다.

현재 많은 전자상거래 시스템에서 XML을 이용하고 있으며, 전자상거래에서 사용되는 문서는 XML을 이용하여 작성하고 전송된다. 그러나 이러한 기업들이 표준화된 XML 문서를 사용하는 것이 아니라 기업의 실정에 맞게 시스템을 구축하여 기업간의 XML 문서 전달에 많은 문제점이 나타나고 있다. 대표적인 예로서 RosettaNet에 따라서 만든 XML 문서를 사용하는 기업과 ebXML(e-business XML)에 따라서 만든 XML 문서를 사용하는 기업간의 거래에 XML 문서를 사용한다면 양자 사이에 XML 문서를 변환하는 작업이 필요하다. 이러한 B2B(Business to Business)에서 양자간에 직접적인 거래를 위해서 변환과 문서 전달 기능을 대행해 주는 것이 XML 허브(XML hub)이다. XML 허브는 XML 문서의 다양성에 따른 엄청난 수의 문서 변환과 HTTP/SMTP 등 다양한 통신 프로토콜로 XML 문서를 주고 받을 수 있는 기능을 제공해 준다[1, 2].

본 논문에서는 XML로 서비스를 요청하고, 서버가 해당 서비스를 처리하는 XML 메시지 처리기를 설계하였다.

## II. 관련 연구 및 기술

### 1. 관련 연구

현재까지 XML을 이용한 메시지 교환 시스템은 기존의 전통적인 EDI(Electronic Data Interchange) 시스템을 어떤 방법으로 XML로 메시지를 교환할 것인가에 초점이 맞추어져 있다. EDI/XML 시스템의 성공적인 구축은 '데이터 교환 모델'을 위하여 XML을 사용하고, '모습을 표현하기 위하여' XSL(XML Style Language)을 이용하며, 전통적인 EDI와 쉬운 통합

방안을 지니기 위하여 DTD를 사용하고, 문서중심의 조회와 처리가 가능케 하며, 타 정보 시스템과의 연동이 가능하도록 개발하는 것이다[2].

이러한 문제를 해결하기 위하여 전자상거래 표준 기술로서 XML이 등장하게 되었으며 광범위한 영역에서 입지를 확고히 해 나가고 있다. 또한 여러 가지 장점을 가지고 있기 때문에 다양한 분야에서 적용이 되고 있다[9].

RosettaNet은 정보기술 및 전자부품의 SCM(Supply Chain Management)을 위한 XML 기반의 비즈니스 표준을 개발하기 위해 1998년에 결성된 컨소시엄으로, 400여 개 이상의 업체가 참여하고 있다. RosettaNet에서는 비즈니스 프로세스를 정의하고 데이터 교환을 위한 기술규격을 제공하고 있다.

RosettaNet에서 정의하고 있는 표준으로는 Dictionary, RNIF(RosettaNet Implementation Framework), PIP(Partner Interface Process)가 있다. RosettaNet Dictionary는 크게 비즈니스 부분과 기술 부분으로 구분된다. Dictionary는 비즈니스에서 사용되는 공통된 용어와 속성들을 표준화한 것으로, 비즈니스를 위한 공통 전자상거래를 위한 XML 메시징 시스템 개발 플랫폼을 제공하여 개별 기업의 중복되는 투자와 노력을 절감하는 역할을 한다. RNIF는 RosettaNet 표준에 대한 시스템의 신속하고 효과적인 개발을 위한 가이드라인과 통신 프로토콜, 보안에 관련된 부분을 명시하고 있다. 즉, XML과 HTML을 사용하여 거래 파트너 사이에 정보를 교환하는 방법을 정의한다. RosettaNet에서 제공하는 표준 중에서 가장 중요한 것은 PIP이며, PIP는 거래 파트너와 인터페이스 할 수 있는 비즈니스 프로세스를 정의하고 있다. PIP는 크게 6개의 클러스터로 구성되어 있으며, 각 클러스터는 다시 세그먼트 단위로 구분되고, 세그먼트 안에 하나의 PIP가 정의된다. RosettaNet에서는 비즈니스 모델, Dictionary, RNIF가 PIP의 입력이 되며, PIP가 거래 당사자들에게 배포되고, 각 기업에서 해당 소프트웨어를 개발하는 로드맵 역할을 한다. 각 PIP는 모든 비즈니스 로직, 메시지 흐름, 메시지 내용을 포함한다[7].

Microsoft사는 1999년 'BizTalk Initiative'라는 이름을 가지고 XML 기반의 B2B 전자상거래 솔루션을 발표하였다. BizTalk은 XML을 이용하여 기업 내부 또는 기업간 응용 프로그램 통합을 효과적으로 할 수 있는 기반을 제공해 줌으로써 보다 빠르게 전자상거래를 구축할 수 있는 방법을 제시하였다. BizTalk Initiative는 BizTalk Framework, BizTalk.org, BizTalk Server의 세 가지 요소로 구성되어 있다. BizTalk을 바탕으로 하는 B2B 전자상거래 시스템에서는 거래자들이 비즈니스 문서들을 교환하기 위하여 BizTalk Server를 이용한다. 기업에서 구매 요청과 같은 비즈니스 이벤트가 발생할 경우 기업의 응용 프로그램은 XML 기반 비즈니스 문서를 작성하기 위하여 이 비즈니스 이벤트에서 참조할 BizTalk 스키마(schema)를 사용한다.

BizTalk은 B2B 전자상거래와 기업 내부나 인터넷을 통하여 다른 기업간 비즈니스 프로세스를 자동화시키는 플랫폼을 제공한다. BizTalk 솔루션을 이용하여 BizTalk 지원 응용 프로그램과 BizTalk Server 시스템을 구축하고, 거래자들 간에 교환되어질 문서의 양식을 작성하는데 필요한 BizTalk 스키마를 결정함으로써 기업들은 전자상거래 비즈니스를 지원할 수 있게 된다[8].

지금까지는 대부분 국소적인 부분에서 XML 기반의 메시지 전송의 표준을 위한 여러 업체의 솔루션들이 등장하였다. 대표적인 예가 다음의 RosettaNet과 BizTalk이며, 이는 XML 메시징의 표준으로 자리잡기 위해 각 기업에서 전폭적으로 지지하는 것들이다. 그러나 이러한 솔루션들은 XML 문서 전달시, 그 기업의 솔루션을 기반으로 해야 한다는 단점이 있다[7, 8]. 이러한 문제를 해결하기 위하여 본 논문에서 제안하는 방식은 기존의 솔루션을 이용하지 않고, 단지 HTTP/SMTP 방식의 메시지 전송을 사용한다.

## 2. XML

XML은 HTML 같은 고정된 형식이 아닌 확장이 가능한 언어이다. HTML은 태그가 한정되어 있는 반면 XML은 문서의 내용에 관련된 태그를 사용자가 직

접 정의할 수 있으며 그 태그를 다른 사람들이 사용하도록 할 수 있다. XML은 본질적으로 다른 언어를 기술하기 위한 언어, 즉 메타언어이다.

XML은 SGML(Standard Generalized Markup Language)의 간략화 된 버전으로, SGML의 실용적인 기능만을 모은 부분 집합이라 할 수 있다. XML은 SGML의 장점과 일반성을 최대한 수용하는 한편, SGML의 특정 부분을 발췌/요약하여 전문가와 일반인 모두 쉽게 배워 웹에서의 구현을 편리하게 할 수 있는 언어이다. XML이 SGML의 부분 집합이므로 SGML과 XML간의 변환이 용이하고 XML 문서를 SGML 애플리케이션에서 그대로 사용할 수 있다.

XML은 인터넷상에서 뿐만 아니라 전자 출판, 의학, 경영, 법률, 판매 자동화, 디지털 도서관, 전자상거래 등에서는 일정한 플랫폼에 매달리지 않고, XML 상에 정보만을 주어 다른 곳과 의사소통을 하기 위해서는 XML이 매우 광범위하게 이용되고 있다[6].

XML은 사람이 이해하기 쉽고 기계가 다루기 쉬운 구조를 제공한다. 또한 W3C의 표준이며, 세계적으로 영향력 있는 많은 업체들로부터 지지를 받고 있다. 고정된 태그가 없으며, 새로운 태그를 필요에 따라 언제든지 만들어 사용할 수 있고, 태그와 속성을 이용해 스키마의 정의 없이도 데이터베이스에 저장할 수 있다.

XML 문서들은 엔티티(entity)라는 저장 단위들로 구성된다. 엔티티는 파싱되는 데이터 또는 파싱되지 않는 데이터 중의 하나를 포함한다. 파싱되는 데이터는 문자(character)들로 구성된다. 이 문자들의 일부는 문자 데이터(character data)가 되고, 또 일부는 마크업(markup)이 된다. 마크업은 XML 문서의 물리적 저장소 배치도 및 논리적 구조에 대한 설명을 부호화한다. XML은 저장소의 배치도 및 논리적 구조를 강제하는 매커니즘을 제공한다.

XML 프로세서(XML processor)라는 소프트웨어 모듈은 XML 문서를 읽어 들어 그 콘텐츠와 구조에 접근할 수 있도록 한다.

XML은 자료를 구분하는 방법에 대한 표준이고, XSL(XML Stylesheet Language)은 구분된 자료를 출력하는 방법에 대한 표준이다. XSL은 일종의 변환

기술로서 XML의 각 필드를 HTML의 어떤 태그로 변환하여 웹 브라우저에 출력할 것인가에 대한 규칙을 정하는 언어이다.

XML 스키마(XML Schema)는 XML 문서의 구조와 콘텐츠를 정의하는 파일을 일컫는 용어이다. DTD(Document Type Definition)도 이러한 스키마의 일종이지만 많은 문제점을 가지고 있었다. DTD와의 가장 큰 차이점은 DTD는 EBNF라는 복잡하고 약간은 어려운 언어로 기술해야 하지만 XML 스키마는 그냥 XML을 사용하여 기술한다는 것이다. 또한 XML 스키마에서는 DTD에서 표현할 수 없었던 각종 데이터 타입과 엘리먼트 재사용 등이 가능하다. 즉 XML 스키마는 DTD를 대폭 확장한 모델로서 XML 문서가 가질 수 있는 엘리먼트 타입, 엘리먼트 간의 관계, 각 엘리먼트가 가질 수 있는 타입에 대해 상세히 정의할 수 있다.

XML 문서는 일반적으로 각 엘리먼트를 트리 구조로 분리하는 파싱 과정을 거쳐야 한다. 파싱한 자료를 트리 구조로 분석, 저장하여 특정 엘리먼트에 대한 접근을 허용하는 모델을 DOM(Document Object Model)이라 한다. DOM에 따르면 XML 문서는 최상위 엘리먼트가 루트 노드가 되어 계층적인 트리 구조로 문서를 분석하게 된다. 트리 구조로 표현되기 때문에 특정 노드에 대한 접근이 자유로워 Random Access Protocol이라고도 불린다. DOM은 W3C의 권고안으로 확정되어 있다.

XML의 구조는 콘텐츠의 의미를 해석할 수 있게 하는 정보를 제공하기 때문에 고효율의 검색엔진이나 지능적인 데이터 마이닝 등에 대한 새로운 가능성을 제시한다. 또한 XML 태그는 내용의 의미를 기술하고 XSL(Extensible Stylesheet Language)은 프리젠테이션 양식을 제공함으로써 XML 문서 하나는 여러 개의 프리젠테이션 문서(XSL)를 가질 수 있으며, XML을 분석하는 XML 파서는 유니코드를 지원한다. XML 문서의 트리 구조는 문서들의 엘리먼트 수준의 비교와 취합을 효율적으로 하게 하여 이미지, 사운드, 비디오 데이터로부터 자바 컴포넌트, 액티브 엑스 등의 컴포넌트까지 다양하게 수용한다. 이러한 XML은

현존하는 모든 데이터 구조를 대체할 수 있으며, 다양한 데이터 포맷을 지원한다[1].

### III. XML 메시지 처리기

#### 1. XML 메시지 처리기의 구성

XML 메시지 처리기는 서로 다른 시스템 간에 특정 기능을 수행하기 위한 수단으로 헤더(header)와 페이로드(payload)로 구성된 메시지를 사용한다. XML 메시징의 기본 개념은 메시지의 헤더를 XML로 만들고, XML 메시지 브로커(message broker)라는 메시징 서버들은 XML로 된 헤더 정보를 읽어서 메시지와 관련된 기능을 수행한다.

[그림 1]은 XML 메시지 처리기의 간단한 구조를 보여준다. 클라이언트가 XML 문서를 만들어 서비스를 요청하면, XML 메시징 서버는 이를 처리하고 응답 메시지를 만들어 클라이언트에 확인을 해 준다.

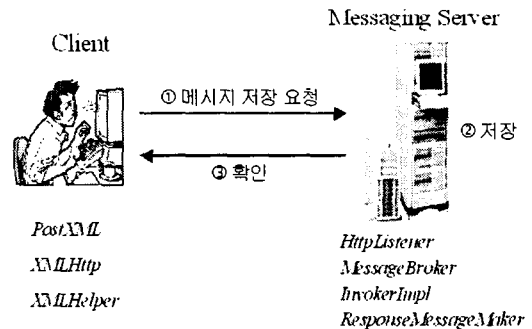


그림 1. XML 메시지 처리기 구조

#### 2. XML 메시지 처리기의 기능

메시지 브로커는 메시징 시스템에서 서버 기능을 수행하며, 받은 메시지에 대하여 헤더 처리, 보안, 에러 및 예외 처리, 라우팅, 호출, 변환의 6가지 기능을 수행한다[4].

##### ▶ 헤더 처리

메시지를 수신했을 때 메시지 브로커에서 가장 먼저

수행되는 기능 가운데 하나이다. 헤더 처리는 수신된 메시지의 헤더 영역을 확인하고, 거기에 포함된 기능 수행을 포함한다. 헤더 처리에 특정 번호를 헤더에 추가하여 추적 가능하게 하거나 헤더 정보가 유효하게 구성됐는지를 검증할 수 있다. XML 메시지 내의 수신자가 적절한지를 처리 과정에서 사전에 검사할 수 있다.

▶ 보안

보안 관점에서 메시지 브로커는 보안의 가장 기본적인 조건인 인증(authentication), 암호화(encryption), 부인 방지(non-repudiation)를 보장해야 한다. 메시지가 수신되면 메시지 브로커는 우선 디렉토리 서비스나 데이터베이스에 저장된 자료로 신원을 확인한다. 해당 자격을 가진 사용자로 확인되면 메시지 브로커는 메시지에 포함된 기능이나 처리에 인증을 받는다.

▶ 에러 및 예외 처리

에러와 예외 처리는 메시지 브로커가 수행하는 중요한 기능 중 하나다. 클라이언트가 수신한 메시지가 유효하지 않거나, 요청을 수행할 수 없는 경우에는 에러 메시지를 클라이언트에 보내야만 한다. 그리고 메시지 브로커 시스템의 문제로 인해 서비스를 제공할 수 없는 경우에도 해당 메시지를 클라이언트에 보낸다.

▶ 라우팅

메시지 라우팅은 두 단계로 이루어진다. 하나는 헤더 라우팅으로, 수신된 메시지가 어느 응용 프로그램에서 처리돼야 할지를 결정한다. 다른 하나는 페이로드 라우팅으로 해당 응용프로그램에서 어떤 프로세스나 메소드가 사용돼야 할지를 결정한다.

▶ 호출

호출 단계에서는 실제 수신 메시지에 있는 페이로드의 자료를 가지고 메소드를 호출하게 된다. 여기서는 메소드 호출을 통해 브로커에서 클라이언트로 반환할 수행 결과가 만들어질 수도 있다.

▶ 변환

변환에서는 다른 형태로의 변환이나 매핑을 수행한

다. 효과적인 변환을 위하여 XSLT(eXtensible Stylesheet Language Transformation)가 많이 사용된다.

## IV. XML 메시지 처리기의 분석 및 설계

### 1. 메시지 처리 순차 다이어그램

사용자는 XML로 헤더와 페이로드를 만들고, 이 XML 문서를 메시징 서버에 전송해서 어떠한 처리를 요구한다. [그림 2]는 클라이언트가 XML 문서를 생성하고, 서버가 클라이언트의 요청을 처리하는 순차 다이어그램을 나타낸 것이다.

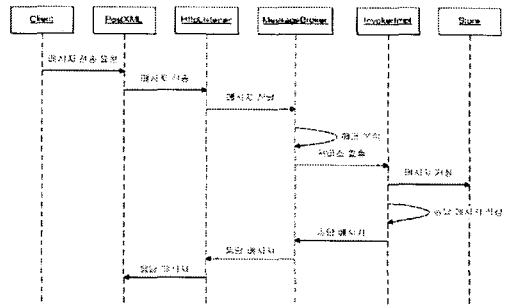


그림 2. 메시지 처리 순차 다이어그램

클라이언트는 사용자 인터페이스를 통하여 요청 서비스를 포함하는 XML 헤더와 페이로드를 만들고, HTTP를 통해 서버에게 문서를 보내 관련된 처리를 요청한다. 서버에서는 HTTP 요청을 받아서, 이를 처리할 MessageBroker에 전달한다. MessageBroker는 먼저 헤더를 분석해서 클라이언트가 요청한 서비스를 확인한다. 그 다음에 해당되는 서비스를 수행하기 위하여 InvokerImpl가 호출되고, 이 클래스는 저장소에 저장한 후, 처리가 성공했음을 알리는 XML 메시지를 만든다.

### 2. XML 메시지 구조

[그림 3]은 XML 메시지 구조의 한 예를 나타낸다. 이 XML 문서는 헤더 부분과 바디 부분으로 나뉘어져 있다. 헤더 부분은 전달할 목적지와 발신자의 정보를

저장하고 있으며, 바디 부분에는 송장(invoice)에 관한 여러 가지 정보를 입력하였다. XML 메시지에서 정보를 저장하는 부분이 바디 부분이므로 이곳에 주문 내역에 대한 정보를 입력하였다.

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="messageList">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="message" type="InvoiceCard"
minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:complexType name="InvoiceCard">
<xsd:sequence>
<xsd:element name="header" type="HeaderPart"/>
<xsd:element name="body" type="Bodypart"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="HeaderPart">
<xsd:sequence>
<xsd:element name="to" type="xsd:string"/>
<xsd:element name="from" type="xsd:string"/>
<xsd:element name="type" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="BodyPart">
<xsd:sequence>
<xsd:element name="order" type="OrderData"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OrderData">
<xsd:sequence>
<xsd:element name="orderNo" type="xsd:string"/>
<xsd:element name="issueDate" type="xsd:string"/>
<xsd:element name="dueDate" type="xsd:string"/>
<xsd:element name="reference" type="xsd:string"/>
<xsd:element name="lineItems" type="xsd:string"/>
<xsd:element name="total" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>

</xsd:schema>
    
```

그림 3. 메시지의 스키마

### 3. XML 메시징 서버

메시징 서버에서의 처리 과정은 [그림 4]와 같다. 클라이언트가 메시지를 만들어서 XML 메시지 서버에 보내게 되면, 메시지 서버는 HTTP로 전송되어 온 메

시지를 수신하는 HttpListener와 이 메시지로 어떤 작업을 처리해야 할지를 결정하는 MessageBroker, 그리고 헤더에 요청된 작업을 실제로 처리하는 Invoker로 이루어진다. 그 이외에 부수적인 Response-MessageMaker가 있다. 그리고 기본적으로 메시지를 파싱하는 XMLHelper와 메시지를 보내고 받는 XMLHttp, 서비스를 요청하는 PostXML의 클래스가 있다.

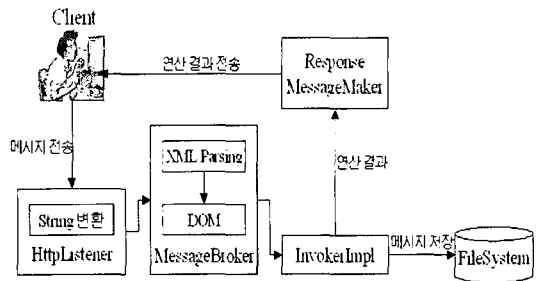


그림 4. 서버 측의 XML 메시지 처리과정

#### ▶ HttpListener

POST 메소드로 넘어온 XML 메시지를 문자열로 MessageBroker에 전달한다. HttpListener는 [그림 4]의 메시지 구조를 이용하여 작성한 XML 문서를 스트림 값으로 받아 Reader로 변환한 후 스트링으로 읽고, 그 코드를 MessageBroker에 넘겨준다.

#### ▶ MessageBroker

HttpListener에서 전달된 값을 파싱하여 DOM을 만들고, invoke 메소드로 DOM을 넘겨서 XML 문서에서 <type> 태그에 있는 텍스트 값을 읽는다.

#### ▶ InvokerImpl

Invoker 인터페이스를 구현한 것이 InvokerImpl 클래스이다. 파일 시스템에 메시지를 저장하고, Response-MessageMaker에서 응답 메시지를 만든다.

#### ▶ ResponseMessageMaker

서비스 요청의 성공이나 실패의 경우에 대한 응답

메시지를 XML 문서로 만든다. InvokerImpl에서 값을 받아 응답 XML 문서를 작성한다.

### V. 결론

기업간의 거래에서 문서를 서로 주고받을 때, 기업의 실정에 맞게 문서를 변형해야 한다. 이 요구를 만족시켜 줄 수 있는 기술 중 하나가 바로 XML이며, 이는 W3C에 의해 표준으로 자리잡았다. 또한 XML은 표준화와 운영체제 중립적이라는 특성 때문에 전자상거래에서 중요한 위치를 차지하며, 이미 많은 전자상거래 시스템에서 이용하고 있다. 이 전자상거래에서 사용되는 문서는 XML을 이용하여 작성되고 전송된다. 그러나 이러한 기업들이 표준화된 XML 문서를 사용하는 것이 아니라 기업의 실정에 맞게 문서를 구축하여 기업간의 XML 문서 전달에 많은 문제점이 나타나고 있다. 이러한 흐름을 반영하여, SCM도 각 기업간의 문서를 전달할 때, XML 문서를 이용하면 효과적이다.

본 논문에서 제시한 XML 메시지 처리기는 기업간 문서 전달의 방식을 자동으로 동작하도록 설계하였다. 현재의 재고상태를 파악하여 거래가 이루어지는 대상 기업에 자동으로 정보를 전송하고 확인하는 방식이다. 이것은 자동적으로 이루어져 특별한 관리 없이 메시지를 전송한다. 이러한 방식은 차후에 SCM 뿐만 아니라, 기존의 B2B 거래에서 유용하게 쓰일 수 있다. XML 메시지 처리기를 B2B 거래 및 SCM을 위하여 적용하면 EDI의 이점과 더불어 기업의 형식에 맞게 변환, 저장할 수 있다. 향후 연구방향으로는 XML 메시징 서비스를 좀더 확장한 XML 허브 시스템을 구현하는 것이다.

### 참고 문헌

[1] E. Xavier, "XML based Security for E-Commerce Applications," Eighth Annual IEEE International Conference and Workshop

on the Engineering of Computer Based Systems, pp.10-17, 2001.

- [2] W3C, Extensible Markup Language(XML), <http://www.w3c.org/XML>, Feb. 1998.
- [3] Alexander Nakhimovsky, Tom Myers, Professional Java XML Programming, WORX, 2000.
- [4] Subrahmanyam Alluamaraju, Professional Java E-Commerce, WORX, 2001.
- [5] KwangMoon Cho, "Packaging Strategies of Multimedia Content in DRM," Proceedings of the 2003 International Conference on Internet Computing (IC '03), pp.243-248, Jun. 2003.
- [6] Elliotte Rusty Harold, W. Scott Means, XML in a Nutshell-A Desktop Quick Reference, O'reilly, 2001.
- [7] RozettaNet, <http://www.rosettanet.org>.
- [8] MicroSoft, "BizTalk Framework 1.0 Independent Document Specification," <http://www.biztalk.org>.
- [9] 신동규, "XML/EDI 시스템의 설계 및 구현", 한국정보처리학회 논문지 8-D권 제2호, 2001.
- [10] 조완수, "UML 객체지향 분석·설계", 홍릉과학출판사, 2000.
- [11] 지영수, "객체지향 Rational Rose 2000", 홍릉과학출판사, 2001.

### 저자 소개

조 광 문(Kwang-Moon Cho)

정회원



- 1988년 2월 : 고려대학교 컴퓨터학과(이학사)
- 1991년 8월 : 고려대학교 컴퓨터학과(이학석사)
- 1995년 2월 : 고려대학교 컴퓨터학과(이학박사)

- 1995년 9월~2000년 2월 : 삼성전자 통신연구소 선임연구원
- 2000년 3월~현재 : 천안대학교 정보통신학부 조교수 <관심분야> : 전자상거래, 콘텐츠 유통, 모바일 콘텐츠, 데이터베이스