

# 컴포넌트 기반의 XML 아답터 설계 및 구현

임동수<sup>o</sup>    박종현    강지훈  
충남대학교 컴퓨터학과  
(lastdkht, jhpark, jhkang)@cs.unknown.ac.kr

## Design and Implementation of a Component-Based XML Adapter

Dong-Soo Lim<sup>o</sup>    Jong-Hyun Park    Ji-Hoon Kang  
Dept. of Computer Science, Chungname University

### 요 약

최근 인터넷상에서 B2B 전자상거래는 상당히 급성장 하고있다. 그러나 기존의 전자상거래 시스템들은 각기 다른 전자상거래 플랫폼과 시스템을 구축하여 사용했기 때문에 전자상거래 시스템간의 서비스 호환과 상호 운영에 많은 어려움이 있다. 이 문제를 해결하기 위해서 전자상거래 시스템들은 국제표준의 공용프레임워크를 준수하고 XML(eXtensible Markup Language) 기반 시스템으로 변화하고 있다. 하지만 기존의 비 XML 전자상거래 시스템과 XML 기반 전자상거래 시스템간의 상거래 정보형식에 대한 상호 호환성은 여전히 문제로 남는다. XML 아답터는 이런 문제점의 해결하기 위해서 XML 기반 전자상거래 시스템과 비 XML 기반 전자상거래 시스템간의 다양한 상거래 문서를 각각의 시스템에서 처리할 수 있는 형식으로 변환하는 기능을 갖는다. 본 논문은 이런 시스템간의 호환성을 해결하고 다양한 상거래문서 형식을 수용할 수 있는 XML 아답터를 컴포넌트 기반으로 설계하고 구현하였다.

### 1. 서론

최근 인터넷이 급속도로 성장하여 기존의 기업들이 현재 갖고 있던 상거래 시장에서 인터넷을 통한 시장 확장을 시도하고있다. 하지만, 기존의 전자상거래 시스템들은 각기 다른 소프트웨어 회사들이 개발한 독자적인 전자상거래 플랫폼들과 구매/판매, 지불, 카탈로그 시스템 등을 구축하여 사용하였기 때문에 전자상거래 시스템 사이의 서비스 호환과 상호 운영에 많은 어려움이 존재한다. 그로 인하여 기존의 전자상거래 시스템을 도입한 회사의 경우 시장 확장에 많은 어려움을 겪게 되고, 점점 폐쇄적인 전자상거래 시장을 형성하게 된다.[3]

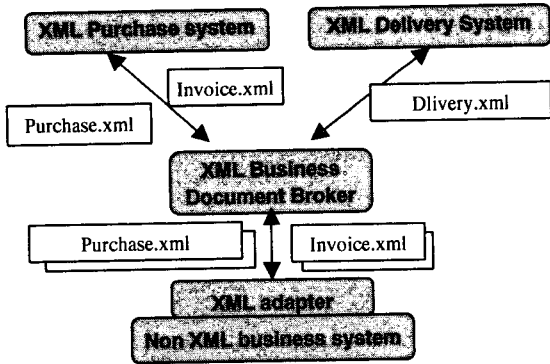
이런 문제점을 해결하기 위하여 CommerceNet 컨소시엄에서는 전자상거래 공용 프레임워크에 관한 국제 표준안인 eCo 규격[7]을 제정하여, 이를 기반으로 전자상거래 시스템을 구축하고 사용하도록 권고하고 있다. 이 프레임워크는 초기에는 CORBA를 기반으로 하였다. 그러나 API중심의 시스템은 통합 방법의 난점과 또 그것에 소요되는 많은 시간과 비용의 문제점을 갖고 있기 때문에, 지금은 웹의 차세대 문서로 자리잡고

있고 loosely coupling integration 방법의 장점을 가지고 있는 XML[4]을 기반으로 하고 있다.

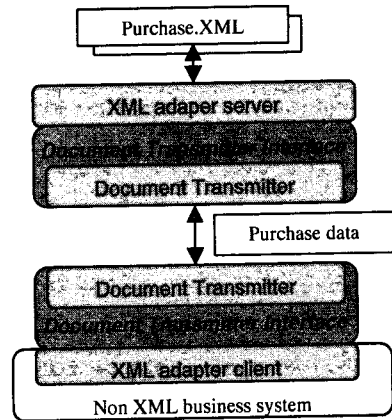
XML 기반 전자상거래 시스템들은 XML문서를 정보의 기본 형식으로 사용하기 때문에 기존의 비 XML 전자상거래 시스템과 상호운영이 불가능하다. 그러나 비 XML 기반 시스템들이 가지고 있는 기존의 시장과 새로이 각광을 받고 있는 XML 기반 시스템을 이용하는 시장을 서로 공유하는 것은 두 시장 모두에게 매우 중요하다.

그림 1은 전자상거래 공용 플랫폼 환경을 나타내며 모든 전자상거래 시스템은 XML 비즈니스 문서 브로커를 통하여 서로 정보를 주고 받는다. 그리고 그림 1과 같이 XML 아답터는 비 XML 전자상거래 시스템 측면에 존재하며, 상거래 정보를 XML 표준 문서로 변환하여 XML 기반 전자상거래 시스템으로 전달하고, 또 그 반대로 XML 기반 시스템의 표준 문서를 특정 비 XML 기반 시스템의 상거래정보 형태로 변환하여 그 목적으로 전달하는 역할을 한다.[1]

\* 본 연구는 전자통신연구소의 지원을 받았다.



[그림 1] 전자상거래 시스템의 상호 운영



[그림 2] XML 아답터의 구성

본 논문은 위에서 설명한 아답터를 기존의 시스템에 적용할 경우, 아답터의 확장성과 재사용성을 높이기 위한 방안으로서 컴포넌트 기반의 시스템 구조를 정의하고, 각 컴포넌트와 XML 아답터 간의 데이터 전송 및 처리를 위한 인터페이스의 정의와 구현에 대해서 설명한다.

본 논문의 구성은 다음과 같다. 2장에서는 문서 변환을 맡는 XML 아답터의 시스템 구조에 대해서, 3장에서는 XML 아답터의 컴포넌트 기반 설계에 대해서 논하였다. 4장에서는 현재 XML 기반 전자상거래 분야에서 진행되는 관련 연구를 살펴보고, 그리고 5장에서는 결론 및 향후 과제를 제시하였다.

## 2. 시스템 구조

### 2.1 XML 아답터

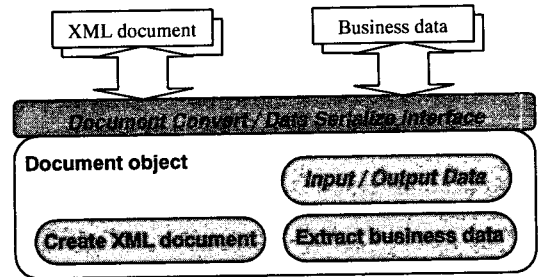
그림 2와 같이 XML 아답터는 크게 3부분으로, XML 아답터 서버와 XML 아답터 클라이언트 그리고 서버와 클라이언트 간의 상거래 데이터 전송을 위한 문서 송수신기로 구성된 다.

XML 아답터 서버는 외부의 XML 전자상거래 시스템과 상거래 문서의 송수신을 담당하고, 전송 받은 XML 문서를 구문 분석하여 상거래 데이터를 추출한다. 그리고, 추출된 데이터를 기반으로 비 XML 전자상거래 시스템으로 전송하기 위한 문서객체를 생성하거나 그 반대로 상거래 문서객체에서 XML 문서를 생성하는 역할을 한다. XML 아답터 클라이언트는 비 XML 기반 전자상거래 시스템이 생성한 문서객체를 XML 아답터 서버로 전송하거나, 그 반대로 문서객체의 상거래 데이터를 비 XML 기반 전자상거래 시스템으로 전송하는 기능을 맡는다. 그리고, 문서 송수신기는 XML 아답터 서버와 클라이언트에서 생성한 문서객체의 상거래 데이터를 전송하는 부분으로 CORBA, DCOM, RMI 등 미들웨어에 독립적으로 처리 가능하게 설계하였다.[1]

위에서 설명한 3부분 중에 본 논문에서는 문서송수신기의 구조에서 상거래 데이터 전송을 위한 인터페이스를 추가하였다. 이 부분이 다양한 상거래 데이터들을 갖는 문서객체가 단일 인터페이스로 다양한 미들웨어를 통해 아답터 서버와 클라이언트로 전송을 가능하게 한다.

### 2.2 문서객체

XML 아답터 서버와 클라이언트 간에 전송되는 구매요구나 송장과 같은 상거래 데이터는 문서객체로 생성되어 그 안에 저장되고, 문서 송수신기에 의해서 전송된다. 그림 3은 문서객체의 구조를 나타낸다. 문서객체의 구성은 문서변환 작업을 위해 상거래 데이터로부터 XML 문서를 생성하는 부분과 XML 상거래 문서를 구문 분석한 후 DOM (Document Object Model)[5][6] 인터페이스를 통해 상거래 데이터를 추출하는 부분이 있다.[1]



[그림 3] 문서객체 구조

XML 아답터 측면에서 다양한 상거래 문서들을 수용하는 것은 급속도로 확장되는 전자상거래 시장에서 중요하다. 이 부분을 만족시키기 위해 모든 문서객체에 대해 아답터가 항상 같은 인터페이스로 접근할 수 있는 문서 변환과 데이터 전송에 대한 두개의 인터페이스를 정의하였다. 그리고 문서 송수신기에서 전송될 상거래 데이터를 생성하고 전달하는 데이터 출력 부분을 추가하였다.

### 3. 컴포넌트 기반 설계

새로운 비 XML 기반 전자상거래 시스템을 쉽게 수용할 수 있도록 XML 아답터를 설계하는 것은 매우 중요하다. XML 아답터를 플랫폼 독립적으로 만들기 위해서 JAVA를 이용하여

컴포넌트 기반 시스템으로 설계하였다. 즉 문서객체는 XML 아답터가 다양한 상거래 문서들을 수용하고 처리할 수 있게 컴포넌트화 하였고, 문서 송수신기는 클라이언트와 서버 사이의 통신이 미들웨어의 종류에 관계없이 독립적으로 이루어지도록 하기 위해서 미들웨어 의존적인 부분을 별도로 컴포넌트화 하였다. 다음은 문서객체와 문서 송수신기의 인터페이스를 정의한 것이다.

### 3.1 문서객체

```
interface IDocObject {
    void CreateData(Document);
    String GenerateXML();
}

interface ISerialize {
    boolean InputDoc();
    boolean OutputDoc();
}
```

[그림 4] 문서객체의 인터페이스

문서객체는 그림 4의 IDocObject와 ISerialize 인터페이스로 구성된다. IDocObject는 XML 아답터에서 상거래 데이터와 XML 상거래문서 간의 변환을 위한 인터페이스로, 모든 문서객체는 이것을 통해 처리된다. 그리고 ISerialize는 문서객체가 갖고 있는 상거래 데이터들을 XML 아답터 서버와 클라이언트 간에 전송을 위한 입출력 인터페이스로, 문서 송수신기는 이것으로 문서객체가 갖는 데이터들의 전송 작업을 할 수 있다. 구매요구서나 송장과 같은 모든 상거래 문서에 대한 문서객체는 위 두개의 인터페이스를 수용하기 때문에, XML 아답터는 효율적으로 다양한 상거래 문서의 변환과 전송에 대한 처리를 지원할 수 있다.

### 3.2 문서 송수신기

```
interface ITransmitter {
    boolean SendDoc(ISerialize);
    void ReceiveDoc(ISerialize);
}
```

[그림 5] 문서송수신기 인터페이스

문서 송수신기는 그림 5의 ITransmitter 인터페이스를 갖는다. 이 인터페이스는 XML 아답터 서버와 클라이언트 간에 상거래 데이터 전송을 할 경우, 다양한 미들웨어를 통해 데이터 전송을 지원하기 위한 것이다. 즉, 문서 송수신기는 미들웨어의 펌프 역할을 하며, 위 인터페이스로 XML 아답터는 문서객체가 갖고 있는 상거래 데이터를 교환할 수 있다.

## 4. 관련연구

전자상거래 문서를 위한 XML 규격에 대한 연구는 비즈니스 문서를 교환하기 위한 Document Framework으로 Commerce

One의 xCBL(Common Business Library)[8]과 B2B (Business-to-Business) eCommerce transaction의 정보들간에 단순한 XML-based protocol을 제공하기 위한 Ariba의 cXML (Commerce XML)[9] 등이 있다.

그리고 폐쇄적인 전자상거래 시장에서 서비스 호환의 문제점과 독자적으로 개발한 전자상거래 시스템간에 상호운영의 어려운 점을 해결하기 위해서 CommerceNet 컨소시엄에서 권고한 eCo 규격을 기반으로, 일반적인 eCommerce Framework을 위한 실재 구조의 설계 및 구현에 대한 연구가 있다.[2]

위 연구들은 eCommerce Framework에 관한 것이라면, 본 논문은 기존의 전자상거래 시스템과 XML 기반의 전자상거래 시스템과의 호환을 위한 상거래 문서의 변환을 다루고 있다.

## 5. 결론 및 향후계획

우리는 본 연구에서 XML 기반 전자상거래 시스템과 비 XML 기반 전자상거래 시스템을 상호 운영할 수 있는 아답터 기술을 개발하였다. 그리고 XML 아답터 소프트웨어의 재사용성과 확장성을 고려하여 플랫폼 독립적이고 컴포넌트 기반의 시스템으로 설계함으로써 다양한 비 XML 전자상거래 시스템에 대해 다양한 상거래문서를 쉽게 수용하고 처리할 수 있도록 하였다.

본 연구의 향후 계획은 XML 아답터를 비 XML 전자상거래 시스템에 적용시키는 방안으로 본 논문이 제시한 서버와 클라이언트로 구성하는 방법과 비 XML 전자상거래 시스템의 컴포넌트로 포함 시키는 방법으로 구분하여 각각에 대한 연구가 필요하다고 생각된다. 그리고, 두 방법에 대한 성능평가와 활용면에서 장단점에 대한 검토가 이루어져야 할 것이다.

## 6. 참고 문헌

- [1] Dong-Soo Lim, Ji-Hoon Kang, Woo-Yong Han, "An XML Adapter for Non XML-based e-Commerce Systems to Interoperate with XML-based e-Commerce Systems," Proc. SCI 2000, Orlando, USA, July, 2000
- [2] Woo-Yong Han, Ho-Sang Ham, "A Method to Design and Implement a XML-Based Common Framework for Internet Commerce," Proc. SCI 2000, Orlando, USA, July, 2000
- [3] B. Meltzer and R. Glushko, "XML and Electronic Commerce: Enabling the Network Economy," ACM SIGMOD, Vol.27, No.4,
- [4] W3C, "Extensible Markup Language (XML) 1.0," www.w3.org/TR/REC-xml, 1998.
- [5] W3C, "Document Object Model (DOM) Level 1," www.w3.org/TR/REC-DOM-Level-1, 1998.
- [6] W3C, "Document Object Model (DOM) Level 2," www.w3.org/TR/DOM-Level-2, 1999.
- [7] CommerceNet, "eCo Architecture for Electronic Commerce Interoperability," eco.commerce.net/rsrceCoSpec.pdf, 1999.
- [8] Commerce One, "xCBL 2.0," www.commerceone.com/xml/cbl, 1999.
- [9] Ariba, "cXML 1.0," http://www.cxml.org/files/cxml.pdf, 1999.