

XMA : XSLT 기법을 이용한 매핑 에이전트

조민양*, 백두권**

*고려대학교 컴퓨터과학기술대학원 소프트웨어공학과

**고려대학교 컴퓨터학과

e-mail : mycho@korea.ac.kr*, baikdk@korea.ac.kr**

XMA : XSLT-based Mapping Agent

Min-Yang Cho*, Doo-Kwon BaiK**

*Dept of Software Engineering, Graduate Schools of Computer Science & Technology, Korea University

**Dept of Computer Engineering, Korea University

요 약

컴퓨터시스템의 발전과 인터넷의 급속한 보급에 따라 인터넷상에서 전자 문서의 표현 및 교환에 대한 요구가 급증하고 있으며, 이러한 수요에 적합한 효율적이고 통일적인 문서 정보의 관리가 이루어질 수 있는 XML 시스템이 각광받고 있지만, 이미 구축되어 원활히 운영되고 있는 레거시(Legacy) 시스템을 일시적으로 새로운 환경의 XML시스템으로 변환하는 것은 기존 시스템을 활용하지 못하고 사장시키는 문제가 있으며, 이를 해결하기 위해, 본 논문에서는 레거시 시스템과 XML 시스템과의 연동을 위해 이식성과 확장성이 높은 XSLT 기법을 이용한 매핑 에이전트(XMA)를 제안하여 레거시와 XML시스템과의 효율적 연동방식을 제공한다.

1. 서론

최근 산업화와 정보화의 발달로 컴퓨터 시스템을 이용한 IT(Information Technology) 산업의 발전은 지속적이며 또한 급속한 발전을 이루어 왔다. 초창기 개별로 진행되던 전산화가 시스템 사용이 성숙됨에 따라 발생된 자료의 상호 교환이 대두되었으며, 이를 위한 많은 방법들이 구현되어 사용되고 있다. 초기의 단순한 파일교환에서 전자문서교환(EDI : Electronic Data Interchange)을 거쳐 최근의 XML(eXtensible Markup Language)에 이르기까지 다양한 종류의 전자적 자료 교환이 이루어지고 있고, 계속 발전되어 가고 있다[1,2,3,8,12].

XML을 이용한 시스템은 자료교환 및 표현에 대한 효율성이 우수하지만, 이미 구축되어 현업에서 기능적 문제없이 활용되고 있는 레거시(Legacy) 시스템들을 단순히 자료교환의 목적으로 XML 시스템으로 변경은 현실적으로 레거시 구축에 소요된 투자 자원의 낭비, 경제적 측면에서 투자 대비 효율성의

저하, 재개발에 따른 개발기간 및 소요비용의 증가, 오랜 기간 축적된 레거시의 기능 누락의 위험성 등 다양한 문제점으로 쉽게 이루어 질 수 없다. 이러한 문제점을 해결하기 위해 본 논문에서는 XML문서의 포맷을 변환하는 XSLT(eXtensible Stylesheet Language Transformation)[6,7,11]를 이용하여 교환 자료의 생성을 담당하는 매핑 에이전트를 제안한다.

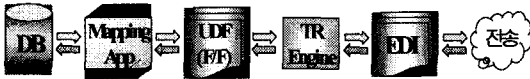
본 논문의 구성은 다음과 같다. 2장에서는 자료교환 관련 연구, 3장에서는 본 논문에서 제안하는 XSLT기법을 이용한 매핑 에이전트(XMA), 4장에서는 XMA 구현 내용 및 기존 시스템과의 비교 평가를 마지막 5장에서는 결론에 대해 기술한다.

2. 관련연구

일반 기업간 비즈니스는 수작업 처리에서 점진적으로 시스템간의 자동 처리되는 전자상거래(EC : Electronic Commerce)로 발전되어 가고 있다. 시스템간 교환을 전제로 하는 전자상거래는 전자문서교

환(EDI)방식에서부터, XML, XML/EDI, ebXML 방식이 있다[1,2,4,5,8,9,10,13].

[그림 1]은 전형적인 EDI 시스템의 자료 교환방식이며, 시스템간 자료교환을 위해서는 외부시스템과 연계하는 부분이 필수적으로 처리되어야 하고, 사용자SW, 매핑SW, EDI변환SW, 송신SW의 구성요소를 갖추어야 한다[8,9].

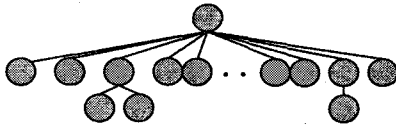


[그림 1] EDI 매핑 프로세스 개요

3. XSLT기법을 이용한 매핑 에이전트의 설계

본 논문에서 제안한 매핑 에이전트는 레거시 시스템에 구축되어 있는 DBMS(데이터 베이스 관리시스템)에서 추출된 자료를 preXML로 변환한 후 XSLT를 이용하여 사용자가 정의한 형태로 변환처리를 하도록 구성되어 있으며, 사용자 종단시스템에서 필요로 하는 형태 즉 비즈니스 포맷으로의 자유로운 변환을 보장한다.

[그림 2]의 preXML은 본 시스템에서 제안한 XML 형태로 스키마 또는 DTD가 필요 없는 Well-formed XML이며, 트리구조가 깊이(depth)가 1 내지 2로 얕고, 폭이 넓은 형태로서 주어진 입력 자료의 항목에 앞뒤로 태그를 붙여 엘리먼트화 시켜 원하는 형태의 변환하는 전처리 단계의 모습이다.

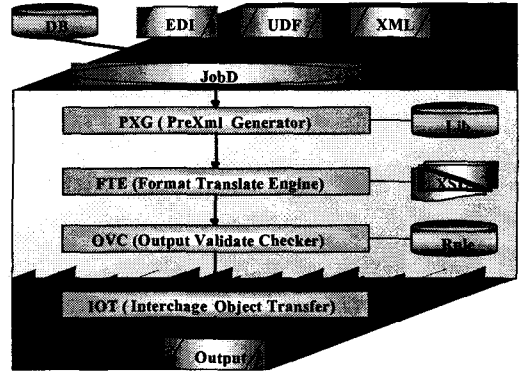


[그림 2] preXML 트리형태

3.1 XMA 구성

[그림 3]은 시스템의 계층적 구조도이고, 입력자료는 EDI, XML, UDF 파일을 수용할 수 있으며, 입력 자료를 감시 확인하여 작업을 분배하는 작업 분배 데몬(JobD : Job Dispatch Daemon), preXML을 생성하기 위한 preXML생성기(PXG : PreXML Generator), XSL과 결합을 통한 새로운 포맷으로 전환을 위한 형태변환엔진(FTE : Format Translate Engine)과 생성된 산출물을 검증하는 산출물검증기(OVC : Output Validate Checker) 및 완성된 결과물을 전송 처리하는 교환객체전송기(IOT : Inter-

change Object Transfer)로 구성되어 있다.



[그림 3] XMA 구조

본 시스템을 계층적으로 세분화하면 3 또는 4 계층으로 나누어 볼 수 있다. 즉 종단 사용자 시스템으로부터 자료를 받기 위한 응용소프트웨어부, 응용소프트웨어와 매핑에이전트의 연계를 책임지는 인터페이스 부분을 포함하여 자료의 형태 변환을 책임지는 매핑에이전트의 엔진부분, 처리 결과를 상대시스템에 전송하는 전송부의 3가지 형태이다. 전송부를 거래상대방에 전달하는 통신단과 데이터베이스로 매핑하는 쿼리실행부로 분리할 경우 4가지 형태로 나눌 수 있으나 크게 3계층을 구분하도록 하겠다. 계층별로 운영 가능한 시스템은 온라인 연결된 웹 시스템과, 오프라인 상태의 기간계 시스템 모두 가능하다. 본 시스템은 다양한 라이브러리와 XSL, 검증규칙을 참조하도록 되어 있는데, 교환정보는 사전에 정의된 서식에 맞추어 생성되어 있어야 한다.

입력 자료는 KEDIFACT 또는 VAN에서 제공된 표준 전자 문서에 기반 하여 제작된 EDI, UDF나 사용자가 정의한 XML 그리고, 기구축되어 있는 기간계 시스템의 데이터베이스의 교환정보가 기록된 테이블로 시작된다.

3.2 XMA 작업 개요

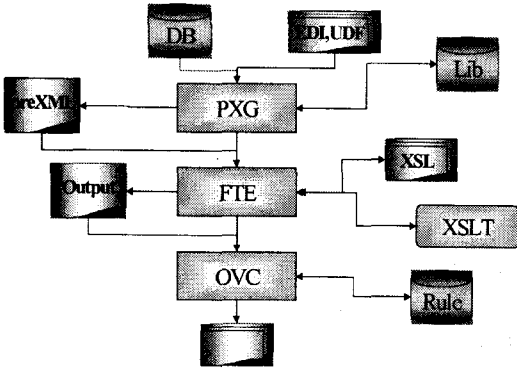
아래 [그림 4]는 XMA의 처리순서도이며 각 계층들의 세부구성은 다음과 같다.

PXG는 입력자료를 판단하여 바이패스(bypass) 형태와 태깅(tagging) 자료를 구분하고 최소한의 파싱(parsing) 정보와 작업 인수를 도출하여 preXML을 생성하여 FTE를 호출한다.

FTE는 넘겨받은 인수를 해독하여 해당하는 XSL

을 선정하여 XSLT를 구동한다. 생성된 산출물의 검증을 위해 OVC를 호출한다.

OVC는 산출물의 구조를 점검하여 이상이 없을 경우 전송처리를 위해 송수신기인 IOT를 호출하며, IOT는 최종산출물을 목적지로 전송처리 한다.



[그림 4] XMA 처리 순서도

[그림 5]는 레거시 DB에서 preXML을 생성하기 위한 알고리즘으로 선택된 DB 테이블에서 필드명과 필드값을 받아들여, 선택자료의 항목 앞뒤로 필드명을 태그로 붙여 엘리먼트화 시키는 것으로, preXML의 생성이 완료된다. 생성된 preXML은 XSLT를 호출하여 지정된 XSL에 의해 최종 산출물인 XML이 생성된다.

```

INPUT
  TBLNAME = 대상 테이블명
  FLDNAME = 대상 필드명
  FLDVALUE = 대상 필드값
START
  Trans_buff = "<" & TBLNAME & ">"
  Print #FN, RTrim(Trans_buff)
  For i = 1 To FLDNAME.Count
    Trans_buff = "<" & FLDNAME(i) & ">" & _
      FLDVALUE(i) & _
      "</" & FLDNAME(i) & ">"
    Print #FN, RTrim(Trans_buff)
  Next
  Trans_buff = "</" & TBLNAME & ">"
  Print #FN, RTrim(Trans_buff)
  Call FTE
END
    
```

[그림 5] DB2preXML 구현 PXG 알고리즘

[그림 6]은 [그림 5]의 DB2preXML 방식에 의한 PXG 알고리즘의 실행 과정 예제이며, 레거시 테이블에서 쿼리 조건에 의한 자료가 선택되면, 선택된 자료의 앞뒤로 필드명을 태그로 처리하여 preXML

을 생성한 경우이다.

	ChungCD	Hochulbh	OrderVsl	Count
1	020	SK12	Tugboat2	12
2	020	SK14	Ulsan	1
3	030	SK12	Tugboat12	6
4	030	HBOOK	HANGJORY	2
5	060	HMM	PANSOLA	4
6	060	SK12	Tugboat12	12
7	300	HCL	FREEDOM12	2

```

Select ChungCD, Hochulbh, OrderVsl, Count
from APTJCT
Where ChungCD = '020'
and Hochulbh = 'SK12'
    
```

```

<APTJCT>
<ChungCD><ChungCD>
<Hochulbh><SK12><Hochulbh>
<OrderVsl><Tugboat2><OrderVsl>
<Count><12><Count>
</APTJCT>
    
```

[그림 6] preXML 처리 실행 예

4. 구현 및 평가

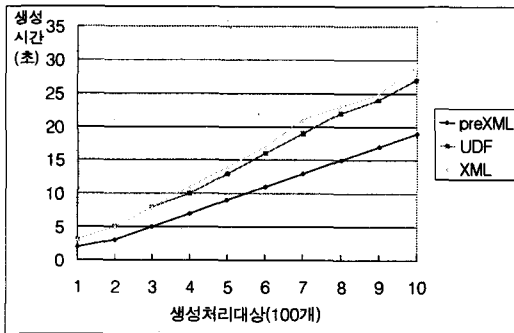
XSLT기법을 이용한 매핑 에이전트는 IBM 호환 PC환경에서 인터페이스 처리를 위해 Visual Basic 6.0과 전송 처리를 위해 Visual C++ 6.0으로 구현되었으며, ActiveX 기술을 적용하여 제작되었기에 Stand Alone, C/S환경 또는 웹 환경에서 사용이 가능하다. XSLT 변환은 공개판으로 PC환경에 빠른 처리 속도를 보여주는 마이크로소프트사의 MSXML 엔진을 사용하였다.

<표 1>은 구현측면에서 시스템간의 특징을 비교한 자료이며, 외부 시스템과 자료 교환을 하지 않았던 레거시, EDI를 통해 교환이 가능한 레거시 및 인터넷 웹 환경에서 사용되는 XML/EDI 또는 XML 시스템과 본 논문에서 제안하는 XMA와 비교해 볼 때, 물론 단순비교가 어렵지만, 기존에 구축된 시스템을 사장시키고 새로운 XML시스템을 도입하는 것에 비해 XMA를 도입하는 것은 관리 측면과 성능측면, 효율성에서 우수함을 나타내었다.

<표 1> 구현측면의 비교

	레거시 (EDI無)	레거시 (EDI有)	XML 신규	XMA
자료교환	불가능	가능	가능	가능
재활용성	불가능	부분가능	불가능	가능
시간	측정불가	XML/EDI 연계레거시 신규시스템 개발에 비해 적은 시간 필요	신규 구축에 따른 재개발로 많은 소요 시간 필요	연계를 위한 부분만 개발하므로 단기간 소요
비용	측정불가	XML/EDI 연계레거시 신규시스템 개발에 비해 저렴한 비용 발생	신규 구축에 따른 재개발 비용 대량 발생	연계를 위한 부분만 개발하므로 저렴한 비용 발생

시스템 성능측면에서 비교를 해보면, 제안된 알고리즘 적용으로 처리 절차 감소를 통한 성능개선을 확인할 수 있으며, [그림 7]은 시스템간 처리성능을 비교한 결과로 기존 시스템에 비해 약 30%정도 처리 속도가 단축된 것을 확인할 수 있다. 기존의 대표적인 전자문서교환 방식인 EDI 환경에 비해 개선된 알고리즘으로 처리 절차의 감소를 통해 성능이 향상되었으며, XML/EDI 방식에 비해 레거시와의 연동이 자유롭고, 역시 처리 절차의 감소를 통해 성능이 향상되었으며, 기존 각 시스템들에 비해 온라인, 오프라인을 구분하지 않고 처리 가능한 효율성의 증가를 볼 수 있다.



[그림 7] 문서 생성 비교도

5. 결론 및 향후연구

본 논문에서는 XSLT기법을 이용한 매핑 에이전트를 설계 및 구현함으로써 XML을 포함하여 EDI문서 및 사용자가 요구하는 다양한 포맷을 처리할 수 있도록 하였다. 또한 문서의 검증을 위하여 KEDIFACT의 표준 전자문서를 이용하여 기존 EDI와의 호환을 위해 검증을 하였다.

본 연구 결과로 인해 그 동안 EDI를 사용해 왔던 기업은 XML을 사용하여 EDI를 확장할 수 있게 되며, 또한 본 시스템을 확장하여 어떤 형태의 전자자료도 교환이 가능한 시스템으로도 이용할 수 있으며 이에 따라 능률적인 시스템 환경 구축이 가능하다.

향후 연구 과제는 첫째 다양한 플랫폼에서 사용이 가능하도록 하여 다른 시스템에서 사용가능 하도록 해야 하며, 둘째로는 교환 대상이 기업의 비즈니스 문서이기 때문에 보안 관련 솔루션을 적용하여 문서를 암호화하여 보낼 수 있는 보안 관련 연구의 접목이 필요하다. 셋째는 현재 기존 EDI와의 매핑을 위하여 스키마(또는 DTD)를 사용하게 되는데 이런 스키마를 자동 생성할 수 있는 시스템 개발에 관한 연

구가 필요하며 생성된 문서나 수신된 문서의 스타일을 변경 및 생성을 위한 스타일 에디터의 개발이 필요하다. 넷째, 이 알고리즘은 좋은 성능을 보이지만, 사용자 정의 포맷을 손쉽게 사용하기 위한 관리들의 사용자 편의성이 부족하다. GUI 환경의 좀 더 친근한 인터페이스를 제공하는 시스템의 보완이 필요하며, 사용자의 요구사항을 통합할 수 있는 기능이 부족하여 사용자의 요구사항에 따라 라이브러리 관리가 늘어나는 단점을 지니고 있다.

참고문헌

- [1] P. Dawkins, "Open Communications Standards : their role in EDI", The Journal of EDI, Vol. 1, 144-152, 1989
- [2] J. Berge, "EDIFACT - a technical introduction", EDI Technology 64-78, 1990
- [3] Martin Bryan, "Guidelines for using XML for Electronic Data Interchange", XML/EDI Group, 1998.1
- [4] David BR Webber, "Introducing XML/EDI Frameworks", 1998
- [5] Frank Boumphreyer, "XML Applications, Wrox Press", 1998
- [6] Michael Kay, "XSLT", Wrox Press, 2000
- [7] Bob Ducharme, "XSLT Quickly", Manning, 2001
- [8] 한국전산원, "SGML, XML, EDI 통합 및 연계방안", 1999.6
- [9] 김태운, "전자거래정보교환(EDI)", 집문당, 1991
- [10] 한국전자거래진흥원, "2000년 EDI-XML 백서", 2000
- [11] W3C, <http://www.w3.org/TR/xslt>
- [12] XML COM, <http://www.xml.com>
- [13] KIEC, <http://www.ebxml.or.kr>