

E-Commerce를 위한 XML문서의 분산화

이상훈*, 송상훈, 노용덕
세종대학교 컴퓨터공학과
e-mail:
taebaik@gce.sejong.ac.kr.

Distributed XML Document for E-Commerce

Lee, Sang-Hun* Song, Sang Hoon.Noh, Yong Deok.
*Dept of Computer Science, Sejong University

요 약

XML은 인터넷을 기반으로 하는 정보교환의 매개체로서 다양한 응용분야로 확산됨에 따라 전자상거래를 위해 최근에 매우 중요한 기준이 되었으며 많은 소프트웨어 디자이너들은 XML 형식의 정보를 토대로 새로운 시스템을 개발하고 있다. 본 논문은 XML을 처리하기 위해 일반적인 아키텍처를 제안하는데 XML과 XSLT와 HTTP와 Java Servlets 같은 기술을 사용하면서 XML 프로세싱 시스템을 디자인 한다. 본 연구는 일반적이면서도 유연하고, scalable하며, 확장성이 있으며 또한 분산 프레임 환경에 적합하게 디자인을 하고있다. 이러한 아키텍처와 시스템의 주요한 응용은 전자 산업 시스템의 데이터 교환을 지원하는 중요한 것이 되었기에 이에 소개하고자 한다.

1. 서론

XML은 데이터를 저장하고 교환하기 위해서 대단히 유연한 형식인데 XML은 특히 전자상거래 응용 프로그램을 위해 웹 응용 개발자로부터 많은 주목을 받고 있으며 XML은 Machine-Architecture 라는 독립적인 형식이기 때문에 다른 데이터를 사용하고 있는 회사끼리의 데이터의 교환을 쉽게 한다.

이 논문에서는 접근방법을 달리하여 XML processing 아키텍처를 디자인했는데 아키텍처의 핵심은 XML 문서가 처리되는 정보를 입력받아 결정하는 문서통합자(Document Integrator)라고 하는 것을 사용하였다. 프로세싱은 높은 수준의 스크립트에 기초를 두고 있다. 또한 DI는 통합자가 XML 문서를 처리하는 스크립트에 기초를 두고 있다.

통합자(Integrator)에 의해 만들어진 XML은 XML 형식의 문서를 통합자에게 되돌려준다. 그때 통합자는 만들어진 문서를 처리하므로서 필요한 다른 모

듈을 만들 수 있다.

2.디자인 목적

시스템의 구조는 다음과 같은 목적을 가지고 있다.

1.일반성- 시스템이 변경 없이 공통적인 XML 프로세싱 요건에 맞게 바꾸어 질 수 있다.

2.모듈화-각 모듈은 하나의 카테고리를 제공하는 것에 책임이 있으며 프로젝트 관리와 소프트웨어 유지보수를 용이하게 한다.

3.분산화 -시스템이 느슨히 결합된 다양한 모듈로 나뉘질 때 다른 모듈이 프로세싱 파워를 향상시키기 위해 다른 컴퓨터에 분산되어 진행된다.

4.확장성-새로운 능력이 새로운 모듈을 더하는 것에

소개되고 새로운 능력을 더할 때 시스템의 코어 부분은 변경이 필요 없다.

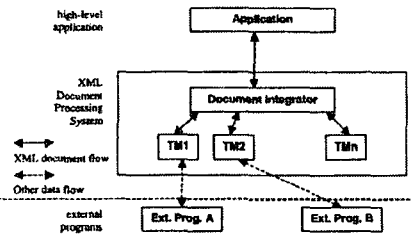
5. 융통성- 프로세싱 논리가 script와 같은 파일에서 실행된다.

3. 시스템 아키텍처

3.1 문서통합자(Document Integrator)

DI(문서통합자)는 응용 프로그램으로부터 XML 데이터를 입력받기 위한 아키텍처의 핵심이다. 네트워크 커백션에서 디스크 파일로 올 수 있는 XML 문서를 받자마자 DI는 응용 프로그래머에 의하여 작성된 화일 스크립트에 따라 입력 문서를 처리한다. 파일이 DI를 적합한 XML 문서에 전해 주면서 다양한 모듈변화가 스크립트에서 설명되며 TM(Transformation Modules)으로부터 받은 문서는 XML 문서인데 DI는 프로세싱을 위해 임시로 저장할 수 있다.

XML 임시파일을 다른 TM에게 줄 수 있으며 TM(Transformation Modules)으로 부터 모든 결과를 수집한 후에 DI는 XSLT에 의해 결과를 결합시키고 최종적인 응용 프로그램에 대한 결과를 XML의 응용 프로그램에 돌려준다.



<그림-1 시스템모델>

DI의 주요한 역할은 데이터를 TM(Transformation Modules)에게 전송하는 역할뿐만 아니라 응용 프로그램과 TM 사이에서 다리 역할을 하는 것인데 TM(Transformation Modules)은 SwitchBox의 문서 역할을 하고 XML 문서의 실제 프로세싱이 다양한 TM(Transformation Modules)에게 보내지고 XML 문서를 TM 사이에 전달하기 위해 DI는 XML 문서의 데이터를 메시지로 보낼 수 있어야 한다. 이것은 XML 문서를 변형시켜야 하는 것을 의미하고 있으며 변화가 TM에게 전달된다. 그러나, 성능의

향상을 위해서는 DI에게 능력을 추가하고 XSLT를 DI에 포함시키면 된다.

3.1.1 XML Transformation (XSLT).

XSLT가 XML로부터 HTML로의 변환을 담당하는 부분으로 화면에 어떻게 보여질 것인지를 결정하는 요소로서 XSLT에 의해 표현되는 변환을 stylesheet라고 부른다. 다시 말하자면 XML로 표현된 source tree를 html 형태의 result tree로 변환하는 것이라고 말할 수 있겠다. 따라서 입력과 중간 수준의 XML 파일을 잘 다루기 위해서 DI 시스템에서 XSLT를 선택 한다.

3.2 Transformation Modules (TM)

각 TM은 하나의 카테고리를 다루는 것이고, 전형적으로 TM은 DI로부터 XML 문서를 받고 TM이 특정한 스크립트 프로그램에 의해 수정된다. 비록 TM이 XML의 응용임에 틀림없지만 DI로 돌아가서 XML 문서와 정확한 형식과 전달된 XML 문서가 TM 프로세싱의 결과로 나타난다. 그리고 TM이 들어온 XML 문서를 처리할 것인가를 결정하는 것은 Implementers 시스템까지이다.

3.2.1. Database Access TM (DATM)

DATM은 보통 관계 데이터베이스에 대한 접근을 제공하는 TM의 한 예인데 XML이 data를 제공하는 서버에 테이블과 필드가 접근하는 지에 대한 정보를 나타내고 있다. Query의 경우에 Query가 XML의 정보를 제공하는 것으로 알려져 있으며 DATM은 XML 문서로 바뀐 후에 DI에게 결과를 묻는다.

3.2.2. Message Generation TM(MGTM).

MGTM(Message Generation TM)이 XML 메시지로서 자료로 제공하는 데이터를 해석하고 메시지 헤더나 다른 적합한 논리에 따라 메시지를 보낸다. 이것은 메시지가 성공적으로 보내졌는지 아닌지 가리키기 위해 XML문서를 DI에게 되돌려 주는 것이다.

MGTM은 간결한 메시지나 이-메일이나 팩스나 Usenet 뉴스그룹이나 호출기 메시지나 프린트나 심지어 이동하기 쉬운 전화 같은 다양한 미디어를 통해 메시지를 보낼 수 있다. 그러나 메시지가 매체의 메시지 형식과 일치해야 한다는 제약이 있다.

메시지를 전송하기 전에 MGTM이 대상 매체의 형식에 들어온 XML 메시지를 바꿀 수 있다. 아마 MGTM이 외부의 프로그램을 통해 메시지를 보내거나 외부용 네트워크는 XML이 아닌 형식을 사용하면서 연결한다.

3.2.3. Em Gateway TM (EmGTM).

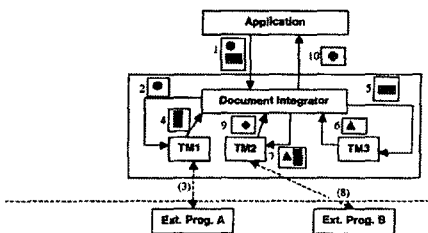
많은 회사들이 메시지를 교환하기 위해 EDI을 사용하고 있으며 EDI-GTM은 송장과 같은 XML 문서나 구매주문서 같은 EDI 메시지를 받아들이고 문서내용을 EDI 형식으로 바꾸고 EDI 채널을 사용하면서 보내는데 이것은 XML문서 프로세싱 시스템과 EDI시스템 사이에 전송되고 있는 메시지를 내보내기 위해 게이트웨이 역할을 한다.

3.2.4. Logging TM (LogTM).

파일시스템에서는 제출된 XML파일을 보내는 TM의 활동을 기록할 수 있으며 스크립트 파일이 XML문서에 있는 내용을 선택하고 LogTM에게 보내기 위해 부분을 수정할 수 있다.

3.3 새 문서

실제 DI가 XML에 의해 만들어진 스크립트 파일을 응용 프로그래머에 의하여 처리하는 것을 의미한다. 프로세싱 논리는 단지 DI가 응용 프로그램을 위한 TM의 XML 파일과 만들어진 XML 파일에 도움을 준다. <그림-2>는 XML이 어떻게 DI를 처리하는가 하는 예를 나타낸 것이다. <그림-2>에서 처음으로 DI는 응용 프로그램으로 XML 메시지-1을 받는다. 이 메시지가 다른 확실한 DI 모양을 스크립트와 <그림-2>에서 보여진 데이터-2 항목을 가지고 있고 처음으로 메시지-2에서 요약되면서 첫째 데이터 항목을 TM-1에게 보낸다.



<그림-2 DI 흐름도>

TM-1이 메시지를 받으면 외부 프로그램이 있는

메시지-3에 영향을 미친다. 그때 새로운 XML 문서가 데이터를 가지고 있으면서 DI에 대한 메시지-4를 되돌려준다. 그리고 메시지-4은 XML 문서로서 Query가 요약된 결과이다.

4. 도구

4.1 DI의 도구

DI의 실행은 make 유틸리티와 개념적으로 유사하다. Make 유틸리티는 Makefile과 유사하며 Makefile이 타겟 목록을 가지고 있다. 타겟이 다른 타겟에 의존할 수 있으며 타겟이 만들어지기 전에 Make는 그 타겟을 의존하고 있는 타겟으로부터 만들어진 스크립트 파일은 DI에게 XML이 중간 수준의 XML 문서를 만드는지 알 수 있다. 따라서 스크립트 파일이 그 자체로 XML 파일이다.

다음 예는 스크립트 파일이 태그를 가지고 어떻게 명시하기 위해 쓰일 수 있는지를 DI의 XML이 증거를 제공하고 입력을 처리하는지 설명한다.

지금 응용이 DI에게 질문을 보낸다고 가정해라:

```
<PRODUCT-QUERY>
<MODEL>IBM-300GL</MODEL>
<MODEL>IBM Intellistation</MODEL>
<PRODUCT QUERY>
```

그때 DI는 루트요소로부터 이 문서 유형의 "PRODUCT-QUERY"을 가지고 있고 그때 이 문서 타입을 위한 적당한 스크립트 파일을 가지고 있을 것을 결정한다. 그리고 스크립트 파일이 위에서 명시한 규칙에 따라 DI가 다음 진행할 것을 가정한다.

XML로 만들어진 형식을 가지고 TM이 데이터베이스를 질문한 결과를 되돌려주는 것이 DATM다. 스크립트 파일에 명시된 데이터베이스 이름과 테이블 이름이 DI에 의하여 TM에게 전해져 진다. TM은 파일을 받는 동시에 DI는 다른 데이터베이스의 다른 테이블위에 입력 문서를 TM servlet에게 보낸다. 다음에 DI는 XML이 증거를 제공하는 2 단계로부터 결과 문서를 연결하고 새로운 문서로 바꿀 XSLT에 적용한다.

4.2 웹서버와 WAP 서버와의 통합.

XML 문서 프로세싱 시스템과 다양한 웹 서비스를 제공하고 있는 웹서버를 통합하기 위해서 여러 가지 가능성이 있다.

데이터를 웹 형태로 만들어 사용자로부터 질문을 받아들이고 XML 문서로 Query 형식을 만들고 XML 문서를 DI에게 보낸다. 그때 DI는 자유롭게 작성된 스크립트 파일에 따라 프로세싱을 하고 스크립트 파일은 결과를 웹서버에게 준다. DI에 의하여 웹서버에게 준 문서가 XML 파일인 것에 주목하지만 사용자에게 적절히 전시될 수 있도록 웹서버가 그것을 다시 형식화해야 할 것이다.

스크립트 파일은 웹서버를 통하여 결과를 사용자에게 되돌려준다. XSL 파일은 XML 결과를 적합한 형태로 변경한다.

5. 분산처리

5.1 분산처리

분산처리는 이 논문의 골격이다. DI와 TM이 HTTP를 통해 연결하고 DI와 TM의 본질적으로 장소와 서버의 플랫폼에 제한을 두지 않는다. 그러나 서버가 TCP/IP를 지원해야 하고 모든 방화벽에 의하여 막히면 안된다.

5.2 Portability

만약 TM이 다른 시스템(데이터베이스, 외부의 서버)과 서로 영향을 준다면 다른 시스템이 플랫폼의 독립을 결정할 것이다. TM의 코드나 플랫폼에 의존한 EDI 게이트웨이 같은 응용 프로그램과 서로 영향을 미치는 경우다.

5.3 성능

성능은 응답 시간이 다른 서버로부터 다중 데이터 검색을 하는데 대해 속도가 떨어진다. 이 아키텍처 유사한 데이터 검색을 처리한다. 또한 HTTP를 RMI로 변경함으로써 향상될 수 있다.

6. 결론 및 향후 방향

전자상거래 시스템에서 XML 문서를 처리할 필요성을 확인했고 특정한 응용을 위해 XML 문서를 다루는데 적극적으로 프로그램을 디자인하고 XML 문서를 다루기 위해 일반적인 아키텍처를 디자인 했다. 또한 프로세스 흐름을 컨트롤하기 위해 문서 통합자(DI)를 사용하지만 대부분의 능력을 다양한 TM에게 보내며 또한 효율적인 관심을 위해 DI가 XML 문서를 XSLT와 합병하고 변형시키는 능력을 가지도록 만들었으며 어떻게 DI와 TM을 실행하는 간단한 예를 보여주었다. 또한 디자인이 유연하고

확장성이 있기 때문에 본 아키텍처가 전자상거래의 많은 시스템에 적합하다고 믿는다. 따라서 본 시스템의 구현 및 성능평가를 통해서 시스템에 관련된 추가적인 연구이외에도 좀더 일반적인 시스템으로 확장하기 위한 연구가 필요하리라 생각한다.

7. References

- [1] Tim Bray, Jean Paoli, C.M Sperberg-McQueen. Extensible Markup Language (XML) 1.0, IO-February-1998. "http://www.w3.org/TR/1998/REC-xml-19980210".
- [2] James Clark, XSL Transformations (XSLT) Version 1.0. 16 November 1999. "http://www.w3.org/TR/1999/REC-This xslt-19991116".
- [3] Dan Connolly, XML Homepage on WJC website, April 1997, "http://www.w3c.org/XML"
- [4] Sun Microsystems. Java Homepage, "http://java.sun.com/products/servlet/".
- [5] Sun Microsystems, Java Servlet API Homepage, "http://java.sun.com/products/servlet/".
- [6] W3C, World Wide Web Consortium (WJC) Homepage, "http://www.w3c.org/".