



XML기반의 전자카탈로그 구현 사례

이 승 호*

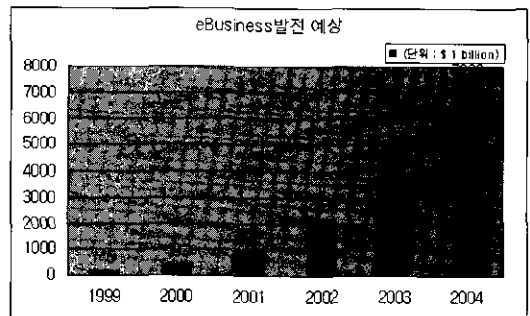
● 목 차 ●

- 1. 서 론
- 2. 구현배경
- 3. XML
- 4. 구 현
- 5. 향후과제
- 6. 결 론

1. 서 론

디지털세상으로 불리워지는 21세기 신경제의 바탕으로 우리는 IT기술의 발전을 지목할 수 있을 것이다. 이는 1990년도말부터 시작된 IT기술의 급성장 말미암아 우리나라를 IT선진국으로 올려놓고 말았다. 이에 대해 우리나라는 주요 기술을 습득하고 있는 것은 사실이나 과연, 이러한 발전발향이 향후 다가올 디지털 글로벌 세상에 어느 정도 대처하고 있는가 하는 것은 다시 생각해 볼 과제인 것이다. 기업에서 현재 진행되고 있는 eBusiness 프로젝트를 중심으로 현재 국내 XML기술의 발전방향에 대하여 생각해 보기로 한다.

이에 따라 국가 차원의 데이터 통합방법을 유도하고 있고, CALs 및 ERP로 연계되는 산업전반적 시스템 구축에 나서고 있다. 이는 향후 기하 급수적으로 증가하는 eBusiness의 성장과도 맞물려서 진행될 것이다.



GartnerGroup: January 26, 2000

2. 구현배경

현재 국내 B2B 및 B2C 다양한 형태의 비즈니스가 진행되고 있다. 이는 IT기술을 기반으로 네트워크사의 MarketPlace를 중심으로 점차 발전하고 있다. 그러나 각 업체간의 데이터 전달에 있어서의 전달 방식 통일의 미비는 그 장애가 되고 있는 것이다.

XML은 기술은 이러한 eBusiness와 밀접하게 관련이 되어있고 향후 비즈니스 거래가 문서통합방법으로서 채택되어 새로운 형태, 현재보다 빠른 eBusiness의 발전을 가져올 것이라고 본다. 그러나 아직까지는 이러한 구현방법의 관련 인프라 구축과 표준화라는 장애로 인하여 IT평균 발전속도에 비하여 더디다는 느낌마저 든다.

* (주)이지시스템 대표이사

3. XML

3.1 등장배경

1990 년대의 WWW(World Wide Web)의 발전은 인터넷 대중화의 가장 큰 기여를 했다고 해도 과언이 아닌데, WWW 의 발전에 가장 큰 공헌을 한 것이 바로 HTML이다. 현재까지 웹에서 가장 많이 사용되고 있는 문서는 HTML로서 하이퍼텍스트(Hyper Text), 하이퍼미디어(Hypermedia)의 기능을 지원하고, 누구나 사용할 수 있을 만큼 간단하며, 특별한 데이터 타입이 사용되지 않고 단순한 텍스트이기 때문에 이식성과 사용이 편리하였다. 이는 HTML DTD에 정의된 태그들만을 이용하여 구현했으므로 사용자는 간단한 태그 만으로도 쉽게 작업할 수 있었다. 단순한 홈페이지 작업을 하기 위해서는 HTML은 매우 편리하였으나 대용량의 데이터 및 텍스트를 관리하기란 어렵고도 불편하였다. 즉 웹에서 사용하고자 하는 데이터(텍스트)의 체계적인 관리 및 사용자의 다양한 요구가 발생하게 되었고 그러한 요구를 수용하려는 시도가 바로 XML 이 등장하게 된 배경이라고 할 수 있겠다.

3.2 소개

3.2.1 구성요소

- DTD (Data Type Definition), Schema
XML문서의 틀을 만드는 부분이다. DTD는 표준으로 제정되었지만 문서 내에서 원하는 부분의 반복 횟수를 지정할 수 없고 제한된 데이터 타입을 제공하여 엄격한 데이터의 검증(Validation)이 불가능하다는 단점이 있다. 그래서 표준화 단체인 W3C에서 XML Schema라는 이름으로 현재 워킹그룹에서 진행 중이다.
- XSL (extensible Stylesheet Language), XSLT (XSL Transformations)
문서의 구조를 정의하고 그 규칙에 따라 XML로 문서를 작성한 후 이를 어떻게보여줄 것인

지를 XSL을 이용하여 스타일시트를 작성한다. XSL은 어떻게 화면에보여줄 것인지를 정의하는 부분(XSL FO:XSL Formatting Object)과 사용자가 원하는 형태로 변환하는 규칙을 정의할 수 있는 부분(XSLT:XSL Transformation)으로 나눌수 있으며 XML의 스타일시트는 XSL를 이용하여 작성되지만 그 내부에서 화면에 보여주는 포매팅 부분은 CSS를 사용하는 방법이 많이 활용되고 있다. XML를 Server Side에서 XSLT 프로세서를 통해 처리하고 웹 서버를 거쳐 사용자에게 전송되는 문서의 HTML과 CSS가 전송되어 웹브라우저에서 처리된다.

3.2.2 XML관련 개념

- Namespace
XML 문서 내의 태그들이 정의된 논리적인 위치를 지정해 주는 역할을 하고 이를 이용하여 태그의 정의 및 태그 식별이 가능하게 된다.
- URI (Uniform, Resource, Identifier)
웹상의 자원들을 나타내는 이름과 주소를 표기하는 구조 형식.
- Xpath
트리구조의 XML에 대하여 트리상의 각 노드를 어떤 경로를 거쳐서 접근할 것인지, 특정 조건을 만족하는 노드만을 추출할 것인지 등에 관련된 표준이다.
- DOM/SAX
XML문서를 처리할 수 있는 기능을 제공하기 위하여 반드시 갖추어야 할 함수들을 IDL이라는 인터페이스 언어로 정의해 둔 표준이 필요하다 이를 위해서 DOM 과 SAX를 사용할 수 있다. 전체문서를 하나의file개념으로 보고, 문서를 구성하는 Tag를 식별하여, 정보를 추출하는 방법이 SAX (Serial Search)이고, XML 문서를 하나의 Tree구조로 분석하여 정보를

추출하는 방법이 DOM (Tree Search)이다.

- RDF (Resource Definition Framework)

인터넷 상에서 메타 데이터를 분배, 교환하기 위한 Framework이다. 즉, 자기가사용하는 MetaData를 RDF의 가이드라인을 따라서 작성하면, 다른 Web에서도 이러한 MetaData를 이해하고, 활용할 수 있다.

한다.

- Indirect Client Side: XML과 XSL을 Client에 넘겨 클라이언트에서 XML과 XSL을 처리하게 한다.

- Server Side : Server에서 XML을 XSL로 변환하여 결과물을 사용자에게 제공한다.

- Cached Server Side : Server에서 자주 반복되는 XSL을 사전에 로딩하여 요청시 해당 XSL을 호출하여 변환 처리한다.

4. 구현

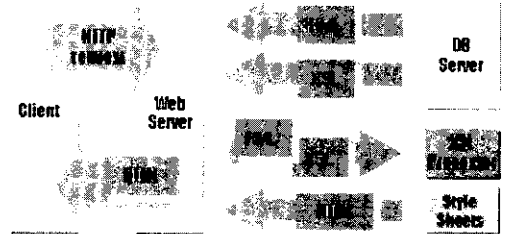
4.1 시스템 개요

4.1.1 Client/Server 역할

XML구현의 기능적 역할에서 Client는 두가지 분류로 정의할 수 있다. 첫째, Thin Client는 고전적인 웹 브라우저의 기능으로서 Server에서 XML이 XSLT와같이 처리되어 HTML형식으로 변환되어 브라우저에 전달하는 것으로 Server에서 처리되는 부분이 많으며 자연스럽게 네트워크 부하가 적은 장점이 있어 많이 사용되어 진다. 이는 Server와 Client 개발자의 역할이 분명하게 구분되어 지고, Server에 주요기능을 집중시킴 으로서 웹 Application의 개발이 매우 용이하다. 둘째, Fat Client는 XML, XSL, DTD, Schema의 자료를 웹브라우저 자체기능을 이용하는 것으로 Server측의 부하를 경감하는 장점이 있지만 상대적으로 브라우저의 기능에 의존하게 되므로 응용 개발의 범위가 빈약할 뿐만 아니라 부하도 증가하게 된다. 하지만 Server 측 개발의 용이함이 있어 구현범위 및 환경에 맞는 Client의 선택이 있어야 하겠다. 하지만 Client의 부하를 최소화하는 관점과 Fat Client의 기능을 지원하는 브라우저의 보급을 및 확장성, 유연성을 고려하면 Thin Client Model을 추천 할 만 하다.

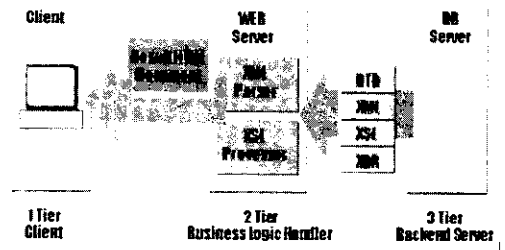
4.1.2 시스템 모델

- Direct Client Side: XML을 바로 제공하고 파싱 변환 등의 작업은 해당 Client에서 처리하게



(그림 1) Thin Client적용한 Cached Server-Side Model

4.3 XML Document 4.3.1 생성 및 처리
Source XML Document(저장된 XML 문서)에서 원하는 자료를 검색하고 추출된 DATA에서 Client에게 보내어지는 HTML문서 까지는 XSL Processor, XML Parser의 처리로 변경하는 작업이 필요하다.



XML을 HTML로 변환하기 위해서는 XSL을 정의하여야 한다. 이를 위해서는 적절한 Template를 구현하는 것이 우선이며 최종적으로 XML을 XSL

에 적용하는 XSL Processor로 HTML 문서를 완성한다.

4.3.2 구현

Source XML Document : 정보제공을 위한 Source XML Document를 작성하기 위해서는 XML Parser의 DOM 또는 SAX API를 활용하여 Processor를 작성하여야 한다. DBMS 제공자는 이를 지원하기 위한 노력이 진행되고 있고 SQL Server 2000의 경우 Web Server(IIS 5.0)의 연동된 SQL Server 가상디렉토리 서비스로 Data를 URL, Template, Xpath를 이용하여 Source XML Document로 변환하여 사용할 수 있는 기능을 제공하고 있고, 이는 웹 사용자에게 XML 자료를 제공할 수 있는 기반을 마련한 것이다.

4.3 Data 추출

4.3.1 Template, XDR Schema, Xpath 활용

Source XML Document에서 Client의 요구에 맞는 Result XML Document로의 변환작업이 필요하고, 이를 위해서 XPath, XDR-Schema 등을 이용하여 구현할 수 있다. XPath는 특정 Elements 또는 Nodes에 대해서 그리고 Parameter와 Variables을 통하여 Data를 추출할 수 있다.

4.3.2 구현

XDR - Schema와 XPath를 이용한 XML자료 작성:

XDR-Schema: DataBase의 View와 같은 개념으로 XML구조의 포맷을 정의한다.

XPath: XDR-Schema의 구조로 생성되는 XML자료에서 특정 Element의 자료를 검색하여 최종 XML 문서를 생성한다.

XDR-Schema 작성: SQL Server 2000과 IIS 서버의 연동으로 XDR-Schema를 지원하게한다.

Schema.xml :

Schema.xml:

```
<?xml version="1.0" ?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes"
  xmlns:sql="urn:schemas-microsoft-com:xml-sql">

  <ElementType name="SPEC_ITEM" sql:relation="SPEC_ITEM">
    <AttributeType name="SEQ" />
    <AttributeType name="CODE" />
    <AttributeType name="VALUE" />

    <attribute type="SEQ" sql:field="PROD_SEQ" />
    <attribute type="CODE" sql:field="ITEM_CODE" />
    <attribute type="VALUE" sql:field="SPEC_VALUE" />
  </ElementType>

  <ElementType name="PROD" sql:relation="PROD_INFO"
    sql:key-fields="PROD_SEQ" >
    <AttributeType name="SEQ" />
    <AttributeType name="KNAME" />
    <AttributeType name="MEMBER" />

    <attribute type="SEQ" sql:field="PROD_SEQ" />
    <attribute type="KNAME" sql:field="PROD_KNAME" />
    <attribute type="MEMBER" sql:field="MEMBER_CODE" />

    <element type="SPEC_ITEM">
      <sql:relationship
        key-relation="PROD_INFO"
        key="PROD_SEQ"
        foreign-relation="SPEC_ITEM"
        foreign-key="PROD_SEQ" />
    </element>
  </ElementType>
</Schema>
```

XPath.xml :

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql">
  <sql:xpath-query mapping-schema="Schema.xml">
    /PROD[SEQ="52355"]
  </sql:xpath-query>
</ROOT>
```

실제 XPath.xml의 XML자료는 Schema에서 정의된 자료에서 SEQ가 52355인 자료가 문서로 생성된다.

4.4 XSLT Processor

XSL 정의 : XML 문서를 HTML로 변환하기 위

해 정의된 Style Sheet 문서.

XSLT Processor : XML문서를 XSL문서에 정의한 대로 실제로 변경하는

Processor로서 XML DOM, SAX(API)를 활용하여 작성할 수 있다.

예) Microsoft XML Version 2.0 DOM(API)를 활용하여 XSLT Processor를 구성한다.

```

<%@ Language=VBScript %>

<html>
<head>
<title>XSLT Processor</title>
<link rel="stylesheet" type="text/css" href="default.css">
</head>
<body>

<%
set source = server.CreateObject("Msxml2.DOMDocument")
source.async = false
source.load("XPath.xml")

set stylesheet = server.CreateObject("Msxml2.DOMDocument")
stylesheet.async = false
stylesheet.load("StyleSheet.xsl")

' Response.Write source.transformNode(stylesheet)
%>

</body>
</html>
    
```

결과, SQL Server Data -> XDR-Schema -> Source XML -> XPath -> Result XML -> XSLT Processor -> HTML로 전개되고, 해당 자료를 요청한 Client는 HTML로 변경된 자료를 받아 볼 수 있게된다.

5. 향후과제

현재 구축되고 있는 XML기반의 B2B사이트는 두각을 드러내는 곳이 없다. 이는 XML의 장점을 공동으로 인식하면 서도 그에 따른 인프라 구축이 요원하다는 것이고, 또한 수많은 기업에서 기간제 시스템과 연동이 되는 B2B Site로 가기를 꺼려한다

는 것이다. 이는 그 유용성을 알면서도 그에 대한 투자비용에 대한 부담이라고 할 수 있을 것이다. 향후 국가차원 또는 민간 차원의 eBusiness를 추진 할 때 표준문서에 대한 인식, 그리고 그에 대한 중심 선도기업이 있어야 할 것이다.

6. 결론

인터넷의 검색기능을 구현하기 위해 데이터베이스에 의존할 경우 보다 XML기반의 검색구조는 웹사이트 구축, 관리에 매우 효율적뿐만 아니라 시스템 Resource 사용측면에서도 적은 비용으로 구현할 수 있다. 이러한 XML 기반의 구축의 최종목표는 인터넷 상에서 MetaData를 분배, 교환에 있다 할 수 있는데, 이를 위해서는 MetaData 공유(RDF : Resource Definition Framework)를 활용 할 필요성이 요구된다. Data관리의 모체 역할을 하는 단체(협회, 기관, 공단 등..)에서 이를 적극 수용하여 표준화 한다면 향후 이러한 데이터 공유로 인터넷의 기능은 새로운 장을 열게 될 것이다.

참고문헌

- [1] 윤홍표, "XML을 이용한 기술정보관리시스템의 구현에 관한 연구", 서울대학교 대학원 석사학위논문, 2000.
- [2] 민만기, "XML에 기반한 소프트웨어 개발 문서 관리 도구", 서울대학교 대학원 석사학위논문, 1999.
- [3] MSDN online XML Developer Center (<http://msdn.microsoft.com/xml/default.asp>)

저지약력



이 승 호

1984년 한국 전자통신 연구원

1990년 (주) 한국에이아이소프트 차장

1992년 (주) 씨엔아이 개발부 이사

1992년-현재 (주)이지시스템 대표이사
(주)한국이지정보 대표이사

관심분야 : EIP, KMS, eCommerce, CRM, IP통신