

# 웹 상에서 XLL의 설계 및 구현

이재동, 김민경  
경남대학교 컴퓨터공학과

## Design and Implementation of XLL on the Web

Jae-Dong Lee, Min-Kyung Kim  
Dept. of Computer Engineering, Kyungnam University

### 요 약

XML은 웹 상에서 문서의 표준화와 데이터베이스로서의 역할을 잘 감당한다. 이러한 이점 때문에 XML 관련 표준안들이 늘어나고 개발되고 있는 추세이다. XML 관련 표준안으로는 XSL, 스키마, 네임스페이스, XLL 등이 있다. 이 중 XLL은 현재 IE(Internet Explorer)5.5에서도 제대로 지원되지 않아 XLL을 구현하기 위한 연구들이 활발히 진행되고 있다. XLL을 지원하는 브라우저, 프로그램, 컴포넌트들이 개발되었는데 이것들은 모두 해당 프로그램을 다운받거나 설치해야 한다는 단점이 있고 특정 프로그램을 사용해야 하는 부담이 따른다. 본 논문은 웹 상에서 아무런 프로그램이 없이 XML문서의 URL을 등록하면 그 문서에 있는 XLL을 DB를 이용하여 저장하고, ASP 및 DOM을 이용하여 DB에 저장된 XLL과 원래 문서에 포함된 XLL을 웹 상에서 바로 보여질 수 있도록 구현한다. XML 문서가 HTML을 넘어서 대중화가 되었을 때 웹 로봇이 XLL 등록하는 역할을 하도록 한다면 XML을 기반으로 하는 검색 사이트로서의 역할을 감당할 수 있다.

### 1. 서론

현재 인터넷은 비약적으로 발전하고 있다. 인터넷이 이렇게 발전하게 되어진 배경은 마크업 언어인 HTML(HyperText Markup Language) 때문이다.

HTML은 SGML(Standard Generalized Markup Language)을 기반으로 하고 있으며, 문서 포맷을 정의하는 표준 텍스트 포맷이다. 하지만 HTML은 태그의 한계성, 단일 링크의 한계성, 정보의 재사용 불가, 지원되지 않는 중첩 구조, 물리적인 구조와 논리적인 구조의 미분리 등의 한계점을 가진다.

SGML 또한 HTML에 비해 복잡하고 웹 상에서는 지원되지 않으며, 애플리케이션 구현 시 많은 어려움이 따른다. 이러한 HTML과 SGML의 단점을 보완하기 위해서 나온 것이 XML(eXtensible Markup Language)이다. XML을 사용함으로써 사용자는 자신이 원하는 태그를 만들 수 있고 그 태그 안에 정보를 넣음으로써 데이터베이스로서의 효율성과 정보의 검색이라는 면에서 뛰어난 효과를 보이게 된다[3].

XML에 관련된 많은 사양들을 IE5.5가 지원을 함에도 불구하고 XLL(eXtensible Linking Language)에 대한 구체적인 지원은 되지 않고 있다[1].

이에 본 논문은 웹 상에서 XML문서의 XLL을 DB로 등록하고 해당 XML문서의 XLL을 보고자 할 때 DB에서 XLL관련 정보들을 찾아 웹 상에서 편리하게 보여줌으로써 XML문서와 관련된 Out-of-Line-Link까지 가능하게 하였다.

### 2. XLL

#### 2.1 XLL 관련 스펙

XLL은 XML에 관련된 모든 링크 부분을 담당하는 요소이다. XLL은 XLink(XML Linking Language), XPointer(XML Pointer Language), XPath(XML Path Language)의 세 부분으로 나뉜다[5, 9].

#### 1)XLink

XLink는 Simple Link와 Extended Link의 두 부분으로 나뉘어 1:1의 링크, 1:다의 링크를 지원한다.

#### ①Simple Link

Simple Link는 1:1의 링크를 뜻하는 것으로서 이제까지 웹을 통해서 많이 접해온 방식중의 하나이다. 하지만 HTML과 다른 점이 있다면 HTML의 A태그에서 제공해주지 못했던 다양한 속성을 제공한다는 것이다[6].

표1. Simple Link의 속성

속성	값	설명
xlink:title		링크하기 위한 이름을 결정
xlink:role		link의 별명을 나타냄. Arc 태그 사용 시, link의 이름을 구별함
xlink:show	replace embed new undefined	화면이 보여질때의 방법을 지정.
xlink:actuate	onRequest onLoad undefined	링크가 언제 실행되는나를 나타냄.

②Extended Link

Extended Link는 1:다의 표현을 하기 위한 링크이다. xlink:type에 표2의 값들을 사용하여 Extended Link에 대한 방식을 결정하게 된다[6].

표2. xlink:type의 값

xlink:type의 값	설명
Locator	위치에 관련된 정보를 나타냄. 양방향 링크를 표현하기 위해 사용됨.
Arc	Locator의 정보 중 원하는 링크만을 골라서 링크시킴.
Resource	링크를 하기위한 자원을 나타냄

2)XPointer

XPointer스펙은 XML문서의 내부구조에 주소를 지정하도록 지원하는 언어에 관해 정의하고 있다. 특히 링크 시키고자 하는 객체가 ID 속성을 가지고 있지 않더라도 이에 대한 pointer를 지정하여 링크시킬 수 있는 특성이 있다. 따라서 ID 속성에 상관없이 문서 내의 엘리먼트, 문자열, 그리고 XML 문서의 다른 부분들에 대한 효율적인 주소 지정 기법을 제공한다[8].

3)XPath

Xpath는 1999년 11월 16일에 Recommendation되었다. XPath는 가장 최근에 추가된 표준안 중 하나로, 원래는 하이퍼링크와 관련해서는 XLink와 XPointer만이 표준으로 존재했다. 하지만 XSL에서 XSLT가 분리되면서 XSLT에서도 문서의 구조를 다뤄야 하므로 XPointer와 XSLT사이에서 이 부분을 통합하는 논의가 진행됐다. 그 결과로 나온것이 XPath이다[7].

2.2 XLL을 구현한 예

Justin Ludwig이 만든 Xlink/Xpointer Tool을 보면 XML을 지원하는 단순한 브라우저이지만, XSL로의

변형, Simple Link와 Extended Link를 지원해준다. DOM, XP, XT 및 SAX API를 사용하여 구현하였다.

ASPointer라는 프로그램은 Stonebroom이 만든 것으로 Windows NT/2000 환경에서만 구동이 가능하다. 이것은 XML, HTML, ASP 등의 파일을 서버에서 액세스하는 업데이트 가능한 Active Server Component이다. W3C의 XPointer draft를 기본 골격으로 하고 있다.

X2X는 Step UK에서 만든 것으로 자바로 구현된 것이다. 링크를 생성, 관리, 유지할 수 있는 XML XLink 엔진으로 External Link(외부 링크)를 문서의 내부로 삽입해서 새로운 XML 문서를 만들어 보여준다. 링크 정보는 ODBC/JDBC에서 저장하여 XML 문서가 인식된 후에 그 문서에 관련된 정보를 뽑아낼 수 있도록 하였다[9].

3. 시스템 설계

XLL의 형식 중 XLink와 XPointer의 두 부분에 대해서 일단 구현을 하겠다. XLink는 Extended Link에 초점을 맞추어서 구현을 하고, XPointer는 어떤 문서의 어떤 엘리먼트를 Resource로 하는 단계까지 표현 하겠다. XPointer 구현 시, 모든 엘리먼트는 id라는 속성을 가진다고 가정하고 id라는 속성을 통해서 XPointer를 구현하겠다.

XLink는 2000년 1월에 W3C에서 발표된 내용을 바탕으로, XPointer는 1999년 12월에 W3C에서 발표된 내용을 바탕으로 구현한다.

본 논문은 마이크로소프트의 IE5.0과 ASP, VBScript(또는 JScript, JavaScript), 그리고 ADO를 결합한 플랫폼에서 작동한다. 마이크로소프트 제품을 선택한 이유는 가장 쉽게 구할 수 있고 IE 자체 내에 XML Parser가 내장되어 있기 때문이다. 그리고 데이터 서비스의 전송 인터페이스는 대중적이고, 거의 모든 플랫폼에서 지원하는 HTTP 프로토콜을 통해서 구현한다.

3.1 시스템 구성

웹 상에서 XLL을 등록하는 화면과 XLL을 보여주는 두 개의 화면으로 구성한다.

클라이언트 쪽에서 XML 문서의 URL을 등록을 하면 ASP, DOM 및 ADO를 사용하여 관련 XLL정보를 SQL DB에 저장한다. 이때 URL을 등록하는 방식은 Post 방식을 사용하며 URL을 등록함과 동시에 URL에 해당하는 문서를 서버 측에 Copy하여 저장한다. 저장한 서버 쪽 문서를 가지고 그곳에 있는 XLL 관련 정보를 DOM을 이용해서 각 속성에 따른 XLL의 값들을 뽑아내어 ASP와 ADO를 이용하여 DB에

저장한다[2, 4].

XLL을 보여주는 화면에서는 해당 DB에서 관련된 정보를 뽑아서 입력되는 XML 문서에 해당 링크 부분을 추가하여 보여준다.

XLL을 등록하는 화면과 XLL을 보여주는 화면을 따로 뚫으므로 인해서 XLL의 Out-of-Line-Link까지 표현할 수 있도록 하였다. XLL을 등록하는 화면을 통해 계속적으로 링크 관련 정보를 등록하고자 하는 XML 문서를 등록하면 된다. 단, XLL을 먼저 등록하고 난 뒤에 XLL을 볼 수 있도록 설계된다.

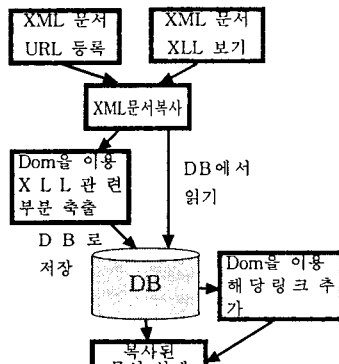


그림1. 시스템 설계

### 3.2 XLink 설계

XLink를 설계함에 Simple Link와 관련된 부분은 서버 측의 XSL문서를 통해서 별다른 조작없이 바로 보여지게 된다. Extended Link의 경우는 Locator와 Arc 관련 부분을 DB로 저장하고 Resource 관련 부분은 Locator가 표현되기 때문에 따로 구현하지 않는다.

#### ①Extended Link 설계

Extended Link를 Simple Link의 형태로 쪼개어서 DB에 저장하고 Simple Link의 형태로 DB에서 문서로 보낸다.

#### • Locator 설계

Locator가 있는 Extended Link를 보면 다음과 같다.

표 3. Locaotr

```

<friend xlink:type="extended">
  <a xlink:type="locator"
    xlink:title="jkimi"
    xlink:href="jkimi.xml"/>
  <b xlink:type="locator"
    xlink:title="yujin"
    xlink:href="yujin.xml"/>
</friend>
    
```

여러 개의 Locator가 있으면 각각에 대한 양방향 링크가 생긴다. 표3의 로케이터는 그림2와 같은 역할을 한다.

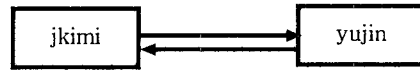


그림 2. locator에 따른 양방향 링크

Extended Link 안에서 Locator가 되기 때문에 Extended Link를 나타내는 태그 안에 있는 xlink:title, xlink:show, xlink:actuate를 가장 먼저 변수에 저장한다. 그 후, Extended Link 안에 있는 Locator의 개수를 뽑아내어 각 개수만큼을 변수에다가 xlink:title, xlink:href, xlink:role을 저장한다. Locator에 의해서 생겨진 각 링크마다 Extended Link에 의해서 생겨진 xlink:show, xlink:actuate 값을 넣어야 한다. 각 Locator마다 양방향 링크가 생기기 때문에 각각에 대한 from 문서와 to 문서가 생길 것이다. 2n만큼의 링크 정보를 생성하여 DB에 저장한다.

표 4. XLL에 대한 DB 정보

Field Name	Type	설명
xlink:from	String	링크가 포함되어야 하는 XML 문서 주소
from:id	"	링크가 포함되어야 하는 XML 문서 주소의 id
xlink:href	"	링크를 할 XML 문서 주소
href:id	"	링크를 할 XML 문서 주소의 id
xlink:title	"	링크시 나타나는 이름
xlink:role	"	XML Link가 가지는 별명
xlink:show	"	보여줄 때 어떻게 보여줄 것인지 결정
xlink:actuate	"	사용자에 의해, 아니면 자동으로 로드될 때 링크를 표현할 것인지 나타냄

#### • Arc 설계

Arc Locator가 포함된 태그 다음에 나오는 태그의 속성으로서 arc를 사용하면 무조건적인 양방향 링크에서 벗어나 자신이 원하는 문서들끼리의 링크만을 표현할 수 있다.

Locator에 관련된 부분을 동일하게 사용하여 변수에 저장을 하되 (xlink:href, xlink:title, xlink:role, xlink:show, xlink:actuate) Arc에서 사용하는 속성인 xlink:from과 xlink:to라는 속성의 값을 가장 먼저 검색하여 꺼낸다.

xlink:from과 xlink:to에 사용되어진 xlink:role 이름을 인식하여 해당 Locator에 있는 정보만 꺼내면 된

다. DB에 저장시에는 xlink:to에 해당하는 값을 xlink:href Field에 넣으면 된다.

• Out-of-Line-Link 설계

일반적으로 out of line link라는 것은 링크정보가 자신의 문서내에 포함되지 않았는데 다른 문서에서 이 링크 정보를 가지고 있는 것을 말한다. 위의 Locator와 arc 설계를 통해서 out of line link가 구현 된다.

3.3 XPointer 설계

XPointer를 설계하기 위해 모든 XML 문서는 id라는 속성을 가진다고 가정한다. id 속성을 통해 모든 엘리먼트는 자신만의 고유한 id 값을 가진다. 이 id를 이용해서 어떤 문서의 특정 엘리먼트가 링크 정보를 포함하고 또 다른 문서의 특정 엘리먼트로 링크 정보가 가도록 하겠다.

DB에서 정보를 뽑아낼 때 from:id라는 필드와 to:id라는 필드가 있다. 이 필드를 통해서 원본 문서의 id 값을 가진 엘리먼트에서 링크로 이어지는 문서에서 해당 id값을 가진 엘리먼트로 이동을 하겠다.

4. 구현

Locator, Arc, XPointer 관련 내용을 DB에 저장하고 난 뒤 보고자 하는 XML 문서에 DB로부터 XLL 내용을 Simple Link로 표현해서 문서에 포함시켰다. XML 문서를 볼 때 링크에 관련되어진 스타일 시트는 따로 지정하여 링크를 볼 수 있도록 하였다.

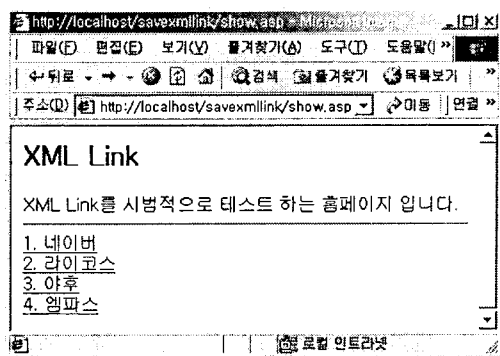


그림 3. XLL을 구현한 결과 화면

5. 결론

XLL을 구현한 많은 시스템 중에서 본 논문은 프로그래밍을 다운받거나 설치하지 않아도 되고, 웹 상에서

구현된 본 사이트에만 접속하면 원하는 XML 문서를 등록하고, 또 그 문서의 링크 정보를 Out-of-Line-Link까지 볼 수 있다는 점이 장점이다.

X2X 시스템은 Out-of-Line-Link에서의 링크를 지원한다는 가정 하에서 XML 문서 안에는 XLL정보가 포함되어 있지 않는다는 것을 전제로 한다. 본 논문은 기존의 링크를 포함한 XML 문서에서도 링크 정보를 꺼내어서 사용할 수 있고 동시에 Out-of-Line-Link를 지원한다. 또, Extended Link를 Simple Link로 변형하여 표현함으로써 Simple Link의 개념에 충실했다.

향후 과제로는 첫째, XLink의 관련 사항뿐 아니라 XPointer의 정확한 스펙에 따라서 다양한 링크 조건을 만족시킬 수 있어야 한다. 둘째, Extended Link의 경우 동적으로 보여주면서(JScript를 사용한 HTML 문서처럼) 각 링크의 목록을 표현해서 Simple Link와 Extended Link의 차이점을 보여주는 것도 좋은 방법이 되어질 것이다. Simple Link로만 모든 Link를 표현하는 것보다 차별화하여 표현하는 것도 앞으로의 과제중 하나이다. 셋째, XLL을 등록하는 부분을 클라이언트에서 관리하는 것이 아니라 웹 로봇을 사용해서 자동으로 XML 문서를 등록하고 XLL을 DB로 저장하도록 한다면 XML에 대한 전문 검색사이트로서의 역할도 할 수 있을 것이다.

[참고문헌]

[1] Alex Homer, XML in IE5 Programmer's Reference, Wrox Press Ltd, 1999  
 [2] Francis Fedorv and etc., PROFESSIONAL ASP 2.0, WROX, 1999  
 [3] Frank Boumphrey and etc., PROFESSIONAL XML APPLICATIONS, WROX, 1999  
 [4] [http://msdn.microsoft.com/xml/reference/cvbrief/XMLDOM\\_Interfaces.asp](http://msdn.microsoft.com/xml/reference/cvbrief/XMLDOM_Interfaces.asp)  
 [5] <http://www.oasis-open.org/cover/xll.html>  
 [6] <http://www.w3.org/TR/xlink>  
 [7] <http://www.w3.org/TR/xpath>  
 [8] <http://www.w3.org/TR/xptr>  
 [9] <http://xmlsoftware.com/xlink>