

DTD 기반의 XForms 자동 생성기 설계 및 구현

김태훈[○] 이은정
 경기대학교 일반대학원 게임웨어전공
 {d75030[○], ejlee}@kyonggi.ac.kr

Design and Implementation of XForms-Generator based on DTD

Kim Tae-Hun[○] Lee Eun-Jung
 Dept. of Computer Science, Univ. Kyonggi

요 약

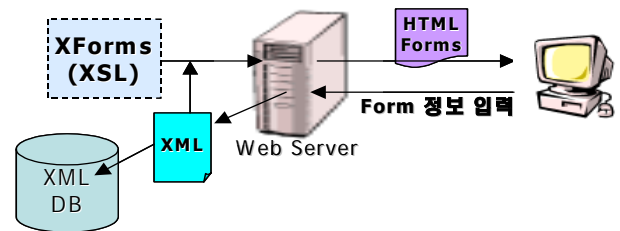
XForms는 XML데이터를 사용하고 내용과 표현을 분리시킨 형태의 Form으로 기존의 HTML이 가지는 특징 외에 쉬운 데이터 처리, 모듈의 재사용성, 다른 장비로의 이식성, 데이터 접근성 등 몇 가지 추가적인 특징을 가진다. 본 논문에서는 XML 데이터의 구조를 나타내는 DTD(Document Type Definition)를 입력받아 XForms를 자동으로 생성해 주는 XForms-Generator의 설계와 구현을 소개한다. 이 시스템은 DTD정보를 분석하여 반복부를 추출하고, 반복부의 처리를 중심으로 XForms코드를 생성한다. 생성된 XForms의 실행 환경은 Orbeon사의 Presentation Server를 이용한다. 개발된 시스템은 XML 데이터를 이용하는 온라인 Form을 개발할 때 시간 및 비용을 줄이는데 도움을 줄 수 있을 것으로 기대된다.

1. 서 론

XML 기술이 점차 확산되어 가고 있는 가운데 온라인 Form에 XML 데이터를 사용하기 위해 XForms 표준이 제시되고 있다[6]. XForms는 기존의 HTML Form이 가지는 특징 외에 XML 데이터의 특성을 바탕으로 추가적인 장점을 가진다. 기존의 HTML Form을 작성하기 위해서는 여러 가지 편집기나 어플리케이션 등을 사용할 수 있다. 하지만, XML 데이터 기반의 웹서버에서 XForms를 이용하려면 직접 XForms 코드를 포함하는 Form 페이지를 작성해야 한다. XForms 기술의 사용이 점차 확대되고 있고, 이에 따라 XForms 코드를 작성하는 방법에 관심이 집중되고 있다.

본 연구의 목표는 DTD 정보를 입력받아 XForms를 자동으로 생성하여 XML 데이터를 이용하는 온라인 Form을 개발할 때 소요되는 비용 및 시간을 줄일 수 있는 시스템을 개발하는 것이다. 이를 위해 본 논문에서는 XForms 코드를 자동으로 생성하는 XForms-Generator의 설계 및 구현을 소개한다. 이 시스템은 DTD 정보를 분석하여 XForms 모델을 생성하고, 반복부를 효과적으로 표현하는 XSL페이지를 구성한다.

본 연구에서 사용하는 XForms 실행 모델(그림 1)은 다음과 같다. XForms-Generator를 통해 생성된 XForms는 웹서버를 통해 HTML 코드로 변환되어 사용자에게 보여지고, 사용자는 Form 정보를 입력하여 submit한다. 입력된 Form 정보는 XML 인스턴스(instance)로 생성되어 XML DB에 저장되거나, 응용될 수 있다. 이 실행 모델에서 사용되는 웹서버는 Orbeon 사의 Orbeon Presentation Server[10]를 사용하였다.



(그림 1) XForms의 실행 모델

본 논문의 구성은 다음과 같다. 2절에서는 XForms, XSL, DTD AST, XForms Renderer 등 관련연구에 대해 간단히 살펴보고, 3절에서는 XForms-Generator에서 사용된 알고리즘에 대해서 설명한다. 마지막으로 4절에서는 결론과 향후 연구 과제를 제시한다.

2. 관련연구

2.1. XForms와 XSL

XForms 표준은 목적, 표현, 결과의 세부분으로 구분되는데, 목적은 XForms Model을 가리키고, 표현은 User Interface를, 결과는 Instance Data, 즉 XML Data를 말한다.

XForms는 XML을 데이터로 사용하기 때문에 XForm에서 입력된 데이터는 이미 well-formed 이다. 또한 저장된 XML Instance Data는 쉽게 응용될 수 있고, 기존에 존재하는 XML 스키마(Schema)를 가지고 쉽게 데이터의 유효성 검사를 할 수 있다. XForms는 Form의 내용과 표현을 분리하여 XForms 모듈의 재사용성, 다른 장치(device)로의 이식성, 데이터의 접근성을 높였다. 또 XForms는 데이터를 처

리할 때 XPath[7]를 기반으로 하여 유효성 검사나 계산 등의 일을 할 수 있다.

XSL은 웹 상에서 XML 데이터의 포매팅 처리를 위하여 제안된 언어이다. XML 문서는 논리적 구조만 가지고 있고, 자체적으로 외부로 보여지기 위한 정보를 가지고 있지 않다. 특정 Element의 폰트가 어떻게 되어야 하는지, 글자색은 어떻게 되어야 하는지 등, XSL은 XML 데이터가 어떤 방법으로 표현되어야 하는지에 대한 내용을 담고 있다. 본 연구에서는 XSL을 이용하여 XForms UI부분을 생성한다.

2.2. DTD 파서와 AST

AST는 DTD를 분석하여 DTD정보를 탐색, 편집할 수 있도록 tree형태로 설계된 자료구조이다. AST는 parser에 의해 파싱(parsing)된 DTD의 정보를 노드로 가진다.

본 연구에서는 DTD 파서(parser)로 ANTLR[9]를 이용하였다. ANTLR 파서 생성기는 파싱 정보에 대해서 정의해 놓은 문법을 입력받아 문법에 따라 DTD 파서를 생성하고 생성된 파서를 이용하여 AST를 생성한다..

2.3. XForms Renderer

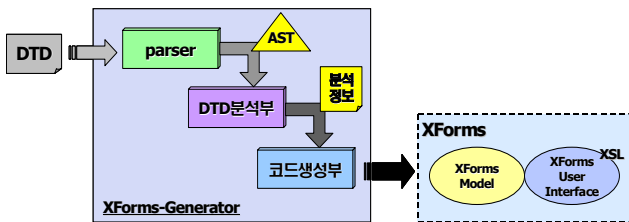
현재 일반적인 웹브라우저에서는 기본적으로 XForms의 렌더링이 지원되지 않고 있다. XForms 코드를 렌더링 하기 위해서는 별도의 XForms Renderer가 필요한데, 오픈소스인 X-Smiles browser[11]와 같은 XForms를 위한 전용 브라우저가 있고, 일반적인 웹브라우저(Microsoft사의 Internet Explorer 6.0)에 플러그인(plug-in) 방식으로 설치되어 작동되는 FormsPlayer[12]와 Novell XForms Explorer[13]등이 있다. 또 XForms를 Server-Side에서 처리하여 클라이언트에게 HTML 형태로 전달해주는 Orbeon사의 Presentation Server[10]가 있다.

본 연구에서 사용하는 Presentation Server는 XForms 모델과 XSL로 구성된 UI(User Interface)를 사용하여 Form을 렌더링하고, Form에 데이터를 입력하고 저장하면 XML 인스턴스(Instance)를 생성한다.

3. XForms-Generator의 설계

3.1. XForms-Generator의 프로세스 및 구성

DTD정보를 기반으로 하여 XForms를 자동으로 생성하는 시스템인 XForms-Generator의 전체적인 프로세스의 흐름은 다음과 같다.

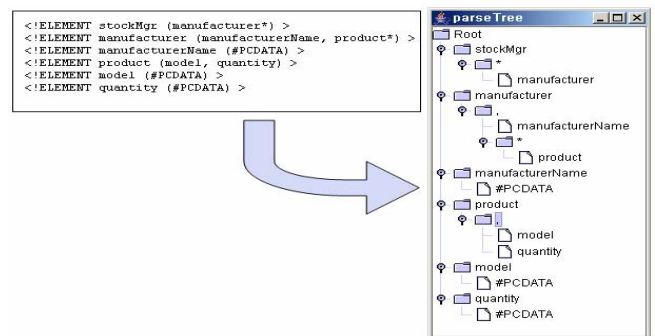


(그림 2) XForms 자동 생성 프로세스

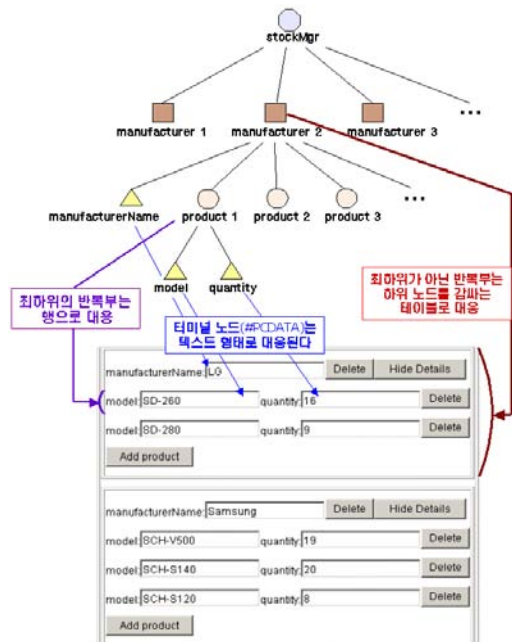
DTD가 입력되면 parser는 DTD를 분석하고, AST(abstract syntax tree)를 생성한다. DTD분석부는 AST를 순회하면서 반복부(*, '+')정보를 수집하고 처리한다. 코드생성부에서는 분석된 tree형태의 자료구조를 바탕으로 XForms를 생성한다. 생성된 XForms 코드는 XML에 대한 정보를 가진 XForms model과 form control을 포함하는 User Interface(XSL)로 구분된다. 이후로는 간단한 예제를 통해 XForms 생성과정에 대해서 설명한다.

3.2. DTD 구조의 분석

DTD분석부는 parser에 의해 분석된 DTD의 정보를 가지고 AST를 생성한다. 다음(그림 3)은 제조사별 핸드폰 제조 목록 관리 DTD를 AST로 나타낸 그림이다.



(그림 3) DTD정보를 통해 생성된 AST



(그림 4) XML 구조와 생성 Form의 설계

(그림 4)는 XML 구조에서 반복부를 표현 방법에 대한 설계이다. 터미널 노드들은 input text 컨트롤로, 반복부는 행과 테이블 형태로 매핑되고 있다. 이후에서는 생성 Form의 표현 규칙에 대해서 설명한다.

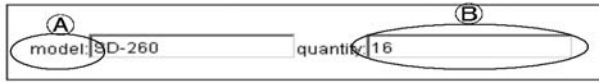
3.3. XForms Code 생성부의 구조

코드생성부는 DTD분석부에 의해 수집된 정보를 바탕으

로 XForms Model과 XForm UI(XSL)로 이루어진 XForms 코드를 생성하는데, 기본적인 코드생성 메커니즘은 다음과 같다.

3.3.1 #PCDATA 필드의 처리

- <label>형태로 변환되는 부분(A)
 - 터미널노드(terminal node)의 Element 이름
- <input>형태로 변환되는 부분(B)
 - 터미널노드(terminal node)의 Element 값 (#PCDATA)



(그림 5) Label과 Input의 Form

```

<xforms:input ref="//model">
<xforms:label>model</xforms:label> (A)
</xforms:input>
<xforms:input ref="//quantity"> (B)
<xforms:label>quantity</xforms:label>
</xforms:input>
    
```

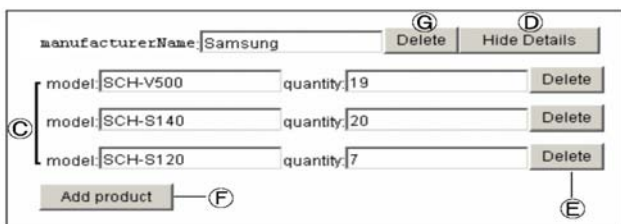
(그림 6) Label과 Input의 code 부분

<label>은 form control로 표현되어야 하는 터미널노드 이름을 표시한다. <input>은 문자열 데이터(#PCDATA, CDATA)를 표현하기 위한 태그이다. XForms 태그의 ref 속성은 표현되어야 할 데이터의 상대경로를 XPath로 나타낸다.

3.3.2 반복부(*)처리

반복부는 <repeat>와 반복되는 노드들을 묶어주는 <bind> (그림8의(H))를 사용한다. <bind>의 nodeset 속성을 이용하여 반복의 대상이 되는 노드들의 노드셋을 만들고, <repeat>에서 생성된 노드셋을 반복 처리한다. 반복부는 생성 Form의 형태(행 또는 테이블)에 대하여 고려해야 하는데, 본 연구에서는 반복부를 표현하기 위해 다음과 같은 규칙을 정의한다.

- 최하위 반복부 데이터 요소들은 열(column)로 표현
- 최하위가 아닌 반복부는 테이블(table)로 표현
- '*'이하의 노드들은 <repeat>태그 블록으로 표현(C)
- 반복되는 부분의 펼침/숨김 버튼(D)
- 반복부 데이터의 해당 요소 삭제버튼(E)/추가버튼(F)
- 반복부 데이터를 바인딩 해주는 <bind>태그(H)



(그림 7) 반복부 데이터를 표현한 Form

XForms Model

```

<xforms:model xmlns:xforms="http://www.w3.org/2002/xforms"
xmlns:xf="http://orbeon.org/oxf/xml/xforms">
  <xforms:instance>
    <stockMgt show_details="" instance_action="" >
      <manufacturer show_details="" >
        <manufacturerName/>
        <product>
          <model/>
          <quantity/>
        </product>
      </manufacturer>
    </stockMgt>
  </xforms:instance>
  <xforms:submission id="submit" method="post"/>
  <xforms:bind nodeset="/form/manufacturer"
    relevant="count(following-sibling:manufacturer)+1 > 0"/>
  <xforms:bind nodeset="/form/manufacturer/product"
    relevant="count(following-sibling:product)+1 > 0"/>
</xforms:model>
                
```

XForms UI

```

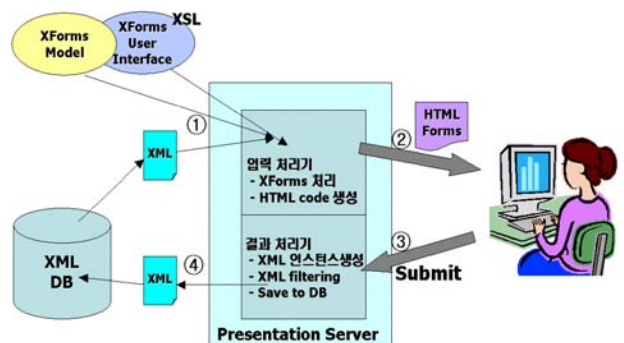
<xforms:submit>
  <xforms:label>
    <xsl:value-of select="if (/stockMgt/manufacturer/Show_details = 'SHOW')
then 'Hide Details' else 'Show Details'"/>
  </xforms:label>
  <xforms:setvalue ref="/stockMgt/manufacturer/Show_details">
    <xsl:value-of select="if (/stockMgt/manufacturer/Show_details = 'SHOW')
then 'NO' else 'SHOW'"/>
  </xforms:setvalue>
</xforms:submit>
</p>
<xsl:if test="/stockMgt/manufacturer/Show_details = 'SHOW'">
  <xforms:repeat nodeset="/stockMgt/manufacturer/product"
    <xp align="center">
      <xforms:input ref="//model"> (A)
      <xforms:input ref="//quantity"> (B)
      <xforms:label>quantity</xforms:label>
      <xforms:input ref="//delete"/>
      <xforms:input ref="//add"/>
      <xforms:label>Delete</xforms:label>
      <xforms:delete nodeset="/stockMgt/manufacturer[index('manufacturerSet')]/product"
        at="index('productSet')"/>
    </xp>
  </xforms:repeat>
  <xsl:if id="E">
    <xforms:input ref="//delete"/>
    <xforms:label>Delete</xforms:label>
    <xforms:delete nodeset="/stockMgt/manufacturer[index('manufacturerSet')]/product"
      at="index('productSet')"/>
  </xsl:if>
  <xsl:if id="F">
    <xforms:input ref="//add"/>
    <xforms:label>Add product</xforms:label>
    <xforms:insert nodeset="/stockMgt/manufacturer[index('manufacturerSet')]/product"
      at="last()" position="after"/>
  </xsl:if>
  <xforms:submit>
    <xforms:label>Add product</xforms:label>
    <xforms:insert nodeset="/stockMgt/manufacturer[index('manufacturerSet')]/product"
      at="last()" position="after"/>
  </xforms:submit>
                
```

(그림 8) 반복부 처리 code

버튼을 생성하기 위하여 사용되는 <submit>은 반복부의 추가/삭제, 반복부의 펼침/숨김 기능에 사용된다. <submit>이 눌러지면 <submit>태그 블록 내부의 XForms action(<insert>, <delete>, <setvalue> 등)을 수행하고 페이지를 갱신한다.

3.4. 생성된 XForms의 실행 환경

다음은 생성된 XForms가 Orbeon Presentation Server[10] 환경에서 실행되는 모델을 나타낸다.



(그림 9) XForms 코드의 실행 모델

Orbeon Presentation Server[10]는 Pageflow라는 기법을 이용하여 XForms를 실행 한다. XForms Model, XForms

UI, 그리고 XForms의 Action을 처리하기 위한 프로세스 등이 정의되어 있다. (1)Orbeon Presentation Server는 생성된 XForms를 입력받아 처리하고 HTML Forms 코드를 생성한다. (2)생성된 HTML형태의 Form이 사용자의 브라우저에 보여진다. (3)사용자는 폼 정보를 입력하고 Submit 한다. (4)입력된 Form 정보를 가지고 XML 인스턴스가 생성되고, 처리를 위해 추가시켰던 속성을 XSLT[8]를 이용하여 삭제하는 XML filtering 작업을 거쳐 DTD에 유효한 문서(validated XML)가 생성된다.

4. 결론 및 향후 연구 과제

본 논문에서는 DTD 정보를 입력받아 반복부를 처리하고, XForms를 생성하는 XForms-Generator를 설계하였다. 다음은 생성된 XForms와 정보를 입력하고 submit하였을 때 생성되는 XML인스턴스를 나타낸 그림이다.

```

<?xml version="1.0" encoding="euc-kr"?>
<stockMgr>
  <manufacturer>
    <manufacturerName>LG</manufacturerName>
    <product>
      <model>SD-260</model>
      <quantity>16</quantity>
    </product>
    <product>
      <model>SD-280</model>
      <quantity>9</quantity>
    </product>
  </manufacturer>
  <manufacturer>
    <manufacturerName>Samsung</manufacturerName>
    <product>
      <model>SCH-V500</model>
      <quantity>19</quantity>
    </product>
    <product>
      <model>SCH-S140</model>
      <quantity>20</quantity>
    </product>
    <product>
      <model>SCH-S120</model>
      <quantity>8</quantity>
    </product>
  </manufacturer>
</stockMgr>

```

(그림 10) XForms 와 XML 인스턴스

XForms Renderer별로 처리방식에 차이가 있기 때문에 생성 XForms 코드에도 차이가 있다. 본 논문에서 구현한

시스템인 XForms-Generator는 Orbeon사의 Presentation Server에서 작동하는 XForms를 생성한다. 이 시스템은 XML 데이터의 생략부, 선택부를 고려하지 않고, 반복부 처리를 중점적으로 하고 있다. 따라서 향후 연구 과제로 생략부, 선택부 처리 기능을 추가하고, 다양한 XForms Renderer에 따른 XForms를 생성할 수 있도록 선택옵션 기능을 추가할 계획이다. 또한 표준화된 XForms를 생성하는 연구가 필요하다.

참고문헌

- [1] 이은정, "XML 반복부 데이터의 변경 협상 방법", 한국정보처리학회 논문지 A, 11권 6호 pp. 0459 ~ 0468 2004. 10
- [2] Cheow, P.Y. and Governatori, G. (2004). "Representing and Reasoning on XForms Document." In Proc. Fifteenth Australasian Database Conference (ADC2004), Dunedin, New Zealand. CRPIT, 27. Schewe, K.-D. and Williams, H. E., Eds., ACS. 141-150.
- [3] John Barton, Tim Kindberg, Hui Dai, Nissanka B. Priyantha, Fahd Al-bin-aliicah, "Sensor-enhanced Mobile Web Clients: an Xforms Approach", Mobile and Media Systems Laboratory HP Laboratories Palo Alto HPL-2003-52, 2003. 03
- [4] 김영철, 강춘길, "XML 어플리케이션을 위한 트리 기반 문서 편집 시스템의 설계 및 구현", 한국정보처리학회 논문지 D, 11권 4호 pp. 0959 ~ 0966, 2004. 08
- [5] 고탁현, 황인준, "Form 기반의 XML 문서 편집기 구현", 한국정보처리학회 논문지 D, 9권 2호 pp. 0267 ~ 0276, 2002. 04
- [6] XForms version 1.1 W3C Working Draft 15 November 2004. <http://www.w3.org/TR/xforms11>
- [7] XPath version 1.0 W3C Recommendation 16 November 1999. <http://www.w3.org/TR/xpath>
- [8] XSLT version 1.0 W3C Recommendation 16 November 1999 <http://www.w3.org/TR/xslt>
- [9] ANTLR. <http://www.antlr.org>
- [10] Orbeon Presentation Server. <http://www.orbeon.com>
- [11] X-Smiles browser. <http://www.xsmiles.org>
- [12] FormsPlayer. <http://www.formsplayer.com>
- [13] Novell XForms Explorer. <http://ftp.novell.com/pub/forge/xforms-explorer/docs/home.html>