

XForms지원 브라우저를 이용한 모바일 오픈 API 플랫폼 개발

유가연, 이은정
경기대학교 정보과학부
e-mail: {despoin, ejlee}@kyonggi.ac.kr

Development of Mobile Open API Platform Using XForms Enabled Browser

Ga-Yeon Yoo, Eun-Jung Lee
Dept of Computer Science, Kyonggi University

요 약

인터넷 환경의 변화와 함께 많은 서비스가 오픈 API 형태로 지원되면서 이들을 결합하고 이용하는 웹 기반의 클라이언트 프로그램이 모바일 환경의 주요한 어플리케이션 형태로 등장했다. 또한 차세대 웹 폼 언어인 XForms는 간편하고 효과적인 UI기술 언어로서 알려져 있으며, 다양한 모달리티를 지원할 수 있고 장치 독립성을 제공할 수 있어 모바일 장치에 더 적합하다.

본 논문에서는 오픈 API를 지원하는 XForms 페이지를 설계하고, XForms의 특징을 살려 성능 및 표현력을 개선한 브라우저를 구현하였다. 또한 오픈 API를 사용하는 모바일 블로그 서비스를 XForms 기반 오픈 API 페이지들을 통해 작성함으로써 제안된 시스템의 표현력과 유연성을 증명하였다.

1. 서론

최근 웹 2.0의 등장을 통한 인터넷 환경의 변화로서 많은 서비스들이 오픈 API형태로 지원되고 사용자가 원하는 형태로 편리한 기능을 제공하는 AJAX 기반의 웹페이지 또는 위젯이 쉽게 구축 될 수 있다.

이에 모바일 환경에서도 오픈 API의 표준화 된 통신방식을 통해 개방형의 서비스 플랫폼을 구축하려는 시도가 계속되고 있다.[10] 그러나 온라인에서 브라우저가 오픈 API를 위한 플랫폼으로 사용하는 것과는 달리 모바일 환경에서는 브라우저가 모바일 단말마다 다르고 통일된 환경이 제공되지 못하기 때문에 브라우저를 플랫폼으로서 이용하는 데 어려움이 있다.

따라서 본 연구에서는 모바일 환경의 오픈 API 플랫폼으로 XForms 기술을 적용하는 방법을 제안하고자 한다. XForms는 XML 데이터를 이용한 웹페이지를 기술하는데 적합하고 서버와의 데이터 송수신에도 특히 강점을 가져 오픈 API를 기술하는데 가장 적합한 언어라고 생각된다.

본 논문에서는 XForms를 모바일 오픈 API 플랫폼으로 사용하기 위한 접근 방법과 브라우저의 구현 모델을 제안하고자 한다. 본 논문의 기여는 다음과 같다. 우선 오픈 API를 지원하는 XForms 페이지의 구조 설계 및 동작 표준을 제시한다. 즉, 스크립트 없이 XForms의 표준요소만으로 모바일 오픈 API가 기술될 수 있음을 보였다. 두 번째로 XForms의 특징을 살려 성능 및 표현력을 개선한 브라우저를 구현하였다. 여기서는 비동기 HTTP통신을 지원하고 콜백을 통한 응답 처리 기능과 액션의 확장된 XML 연산 기능을 추가하였다. 세 번째로 응용 사례 개발을 통

해 모바일 오픈 API를 통해 모바일 한줄 블로그 어플리케이션이 클라이언트와 서버 양측에 높은 유연성을 제공하며 매우 간단하게 작성될 수 있음을 보였다.

2. 관련연구

기존 XForms를 지원하는 플랫폼으로서 온라인 브라우저에서 사용할 수 있는 것은 서버에서 HTML과 자바스크립트로 변환하는 Orbeon 솔루션과 브라우저 플러그인으로서 사용되는 FormsPlayer, 그리고 FireFox나 Opera 브라우저 등 다양하다. 모바일 환경에서는 헬싱키 대학에서 기존의 X-SMILES 브라우저에 XForms를 확장 구현한 사례가 있고[4, 5] 상용화된 서비스 플랫폼을 제공하는 picroforms사의 상용솔루션이 있다[11].

그러나 현재까지 구현된 XForms 구현체들은 클라이언트 페이지의 폼 기능을 구현하는데 목적을 두고 있어 서버와의 상호작용 기능은 중점적으로 다루어지지 않았다. 특히나 모바일 구현체 들은 HTTP 통신을 가정하고 있어 본 논문에서 다루는 오픈 API호출과 처리는 담당하지 못하고 있다.

3. 오픈 API 지원 XForms 페이지 설계

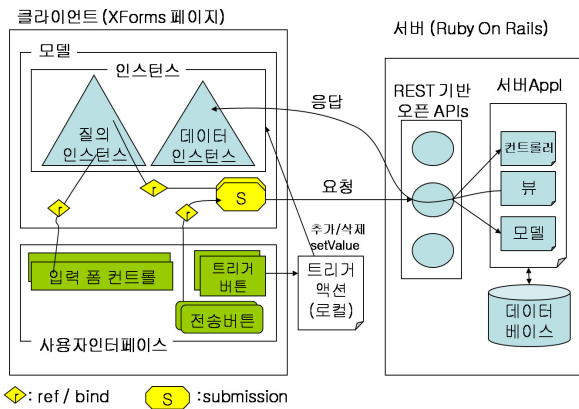
이번에는 XForms 페이지를 이용하여 오픈 API 기능을 제공하는 웹페이지의 구성 방법을 살펴본다. 오픈 API 호출 웹페이지는 서버와의 통신을 담당하면서 요청을 생성하고 응답을 처리할 수 있어야 하는데 아래에서 살펴본듯이 XForms 언어는 별도의 자바스크립트나 프로그래밍 코드 없이 이러한 기능을 기술할 수 있다. 본 논문에서는

XForms 페이지의 예로 모바일 한줄 블로그 어플리케이션을 이용한다. 한줄 블로그 어플리케이션은 모바일 환경에서 간단한 글을 블로그 서버에 추가/삭제/수정 할 수 있고 검색을 통해 게시글을 확인하는 기능도 제공한다.

오픈 API는 흔히 REST방식의 인터페이스를 통해 CRUD(Create, Read, Update, Delete)연산을 제공한다. 본 연구에서는 Ruby on Rails 프레임워크[9]를 사용하는 웹 서버로 모바일 블로그를 위한 REST방식의 인터페이스를 구축하였다.

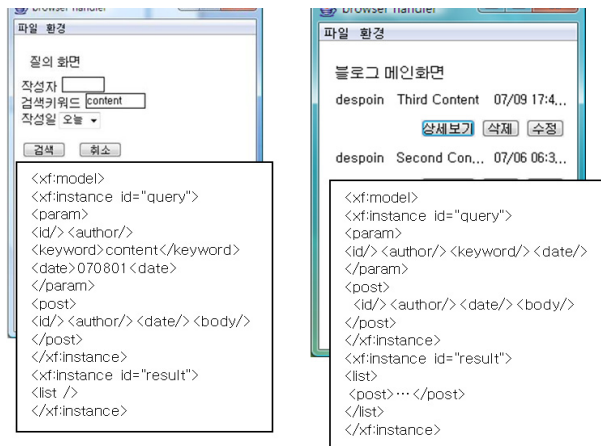
3.1 XForms기반 페이지의 구성

오픈 API를 지원하는 XForms기반의 웹 페이지는 아래 (그림 1)과 같은 구조로 구성된다.



(그림 1) XForms기반 웹 페이지와 서버

내부적으로 XForms 페이지는 인스턴스와 모델, 사용자 인터페이스의 세 개 부분으로 구성된다. 인스턴스는 이 페이지에서 사용할 XML 데이터를 가지는 부분이다. 제안되는 오픈 API 페이지에서는 질의 요청 인스턴스와 결과 데이터 모델 인스턴스를 분리한다. 모델 부분은 XML 데이터 요소에 대한 참조 정보(ref 또는 bind) 및 서버와의 연결 정보(submission)를 가진다. 사용자 인터페이스는 폼이 어떻게 보여지고 사용자 및 모델 인스턴스 데이터와 상호 작용할 것인가를 기술한다. 그리고 버튼과 같은 형태



(a) 질의 메시지 전송 전 (b) 메시지 전송 후 리턴 메시지 반영

(그림 2) 모바일 블로그 뷰와 인스턴스 트리

로 액션을 호출하여 인스턴스 XML 데이터의 수정이나 변경이 가능하다.

(그림 2)는 이하에서 사용할 모바일 블로그의 예에서 인스턴스 트리의 예를 보여준다. 질의 인스턴스는 검색 매개 변수 <param> 부분과 추가/수정을 위한 임시 데이터 <post>를 가진다. 결과 인스턴스는 클라이언트가 서버로부터 받은 결과를 저장하는 트리 <post> 여러 개를 가지고 있다. 이것은 서버가 가진 데이터의 로컬 복제 객체라 할 수 있다.

3.2 submission 코드 구성

오픈 API 호출을 위해서는 미리 약속된 매개변수를 포함한 요청 메시지를 생성하고 서버로부터 회신된 결과 데이터를 클라이언트 페이지에 반영하여야 한다. 이상의 과정을 단계로 나누어 보면 다음과 같다

- ① 사용자가 입력한 정보로 요청의 입력 매개변수 값을 질의 인스턴스에 채운다.
- ② 요청 메시지 생성 단계 질의 인스턴스에서 입력 매개변수를 가져온다.
- ③ 서버로부터 성공 여부와 함께 결과 데이터가 돌아온다.
- ④ 결과 처리 단계에서 응답 결과를 결과 인스턴스에 반영하여 업데이트한다.
- ⑤ 화면을 갱신하여 결과를 사용자에게 보여준다.

여기서 ①, ⑤ 단계는 사용자 인터페이스에 기술되고 ②, ④ 단계는 <xforms:submission>요소에 의해 기술되어야 한다. 요청 메시지의 생성과 응답 결과의 처리는 서버에서 제공하는 통신 방식을 따라야 하고 사용자 인터페이스의 필요사항을 반영하도록 유연하게 구현되어야 한다. 여기서는 한줄 블로그의 예를 통해 CRUD 연산을 수행하는 XForms 페이지가 이러한 요구사항을 반영할 수 있으며 다양한 시나리오의 사용자 인터페이스의 흐름을 유연하게 지원할 수 있음을 살펴보겠다.

먼저 검색에 의한 읽기 연산의 <xforms:submission>요소 기술 예는 다음과 같다.

```
<xforms:submission id="search" method="get"
  ref="instance('query')/param/"
  replace="instance" instance="instance('result')"
  action="http://localhost:9000/search.xml">
  <xforms:setfocus control="mainCard" />
</xforms:submission>
```

검색 기능을 요청하는 <xforms:submission>에서는 요청 메소드가 search이고 get을 이용하는 호출입을 id와 method 속성에서 표현하고 있다. 또한 매개변수로 'query' 인스턴스의 <param>이하의 부분트리를 보내는 것이 ref 속성에 기술된다.

또한 응답 결과가 돌아오면 이것을 'result' 인스턴스에 그대로 대치하면 된다는 것을 replace와 instance 속성이 표시한다. 검색 요청의 결과를 처리한 후 'result' 인스턴스는 회신된 결과를 가지게 된다. 이것은 리스트에 반영되어 해당 블로그 리스트가 화면에 보여지게 된다.

```
<xforms:submission id="create" method="post"
  ref="instance('query')/post/"
  replace="instance" instance="instance('query')"
  target="/post/id"
  action="http://localhost:9000/create_post.xml">
  <xforms:insert origin="instance('query')/post"
    nodeset="instance('result')/list/post"
    at="last()" position="after" />
  <xforms:setfocus control="maincard" />
</xforms:submission>
```

두 번째 예로서 새로운 데이터를 서버에 추가하는 요청의 <xforms:submission>은 위와 같다. 추가 요청을 위해서는 사용자가 폼에 입력한 추가할 게시물의 정보가 'query' 인스턴스의 <post> 부분에 기록된다. 오픈 API 요청 메시지의 생성을 위해 <xforms:submission>에서는 호출 메소드가 create라는 id를 가지고 method가 post 방식임을 기술하였다.

한편 입력 매개변수로 instance('query')/post 부분트리를 전송한다. 응답의 형태는 여러 가지가 가능한데, 성공/실패만 회신되는 경우와 서버에서 부여한 아이디 등의 데이터가 반환 값으로 보내지는 경우, 그리고 추가된 객체 데이터 전체가 내려오는 경우로 나누어볼 수 있다.

본 논문에서는 <xforms:submission>안에 액션을 자식 요소로 넣어서 응답 결과를 처리하는 과정을 기술하게 하였는데, 위의 예에서처럼 결과로 회신된 xml 데이터를 인스턴스의 해당 부분에 삽입(insert) 또는 대치(copy)하는 것이 가능하다. <xforms:setfocus>는 여러 개의 화면으로 이루어진 페이지에서 뷰의 이동을 표현하는 방법으로 제안되었는데 이에 대해서는 아래 절에서 살펴본다.

3.3 카드를 이용한 디스플레이 구성

XForms 페이지가 모바일 오픈 API 플랫폼으로 사용된다면 다양한 서버의 기능을 하나의 페이지에 모은 매쉬업을 제공할 수 있어야 한다. 모바일 환경의 특성 상 여러 가지 기능을 하나의 페이지로 묶은 경우 이것을 나누어서 디스플레이할 수 있어야 하는데, XForms에서는 플랫폼 독립성을 보장하기 위해 이러한 일을 브라우저에게 맡긴다.

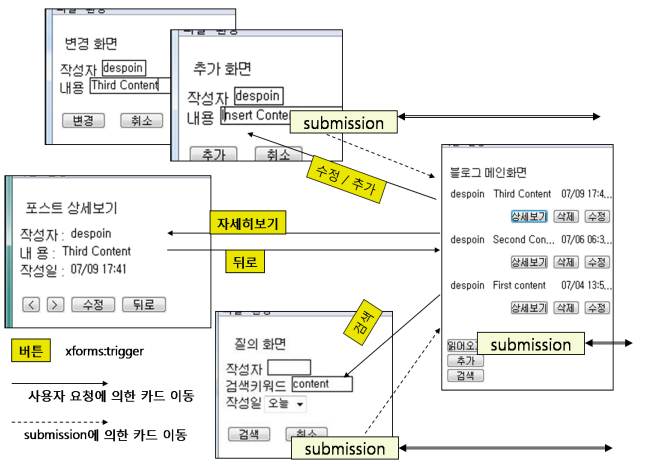
본 연구에서는 한 페이지를 여러 개의 카드로 구성하는 모델을 가정하고 카드 간 이동은 사용자의 행위에 의한 요청이나 해당 작업의 완료 이벤트 등에 의해 발생한다. 즉, 버튼에 의한 다른 카드로의 이동이나 submission의 요청이 완료된 후 원래 카드로 돌아가는 것을 사용자가 XForms 페이지 상에서 기술할 수 있어야 한다. 아래 (그림 3)은 한줄 블로그 예제에 대한 사용자 화면 시나리오이다.

본 연구에서는 이러한 카드 간의 이동과 디스플레이를 사용자가 기술할 수 있게 하기 위하여 XForms에 카드의 개념을 확장하였다.

우선 사용자 인터페이스의 <xforms:group> 요소는 id를 가져서 그것이 해당 카드의 이름이 되고 필요한 경우 별도의 뷰로 렌더링 될 수 있다. 그리고 사용자가 버튼이나 submission의 결과에 의해 카드를 이동하여야 하는 경

우를 표현하기 위하여 <xforms:setfocus>를 사용하였다. XForms 표준의 <xforms:setfocus>액션은 컨트롤의 id를 속성으로 받아 포커스를 이동시킬 수 있다. 여기서는 <xforms:setfocus>의 control 속성에 <xforms:group>의 아이디를 지정하여 포커스의 이동을 통해 디스플레이될 카드를 선택할 수 있게 하였다.

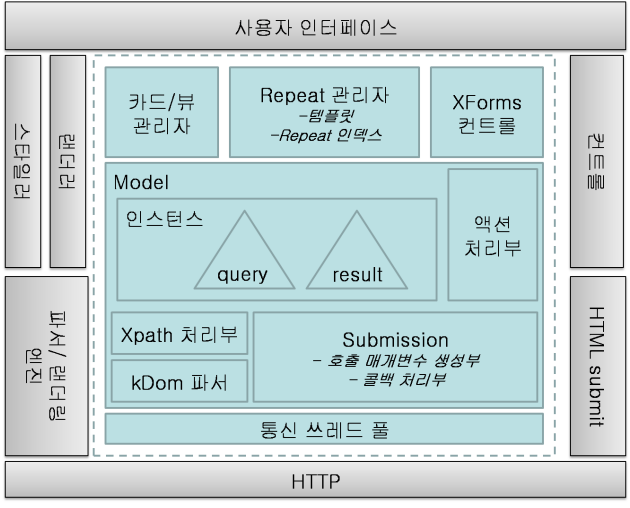
본 연구에서는 위피 기반의 GUI 모델을 사용하는데 여기서는 카드가 스택 방식으로 관리되어야 한다. 그래서 <xforms:setfocus>에 의한 카드 이동 시 이미 스택에 활성화되어 있는 카드라면 해당 페이지까지 돌아가면서 카드를 스택에서 꺼내고 활성화되어 있지 않은 카드라면 푸시하는 방법을 사용한다.



(그림 3) 한줄 블로그 시나리오

4. XForms 지원 모바일 브라우저 설계

본 연구에서는 XForms 지원 모바일 브라우저를 개발하기 위하여 본 연구팀에서 기개발하여 보유하고 있는 위피 환경의 모바일 브라우저인 Xeni Browser를 기반으로 확장하였다. Xeni 모바일 브라우저는 인터넷 상의 XHTML와 CSS로 기술된 페이지를 읽어 화면에 렌더링하고 링크 네비게이션이나 폼 컨트롤을 통한 사용자 입력을 받아 그에 맞게 동작하는 프로그램이다.



(그림 4) 전체 시스템 모듈

본 연구에서는 XForms 확장부를 위의 (그림 4)와 같이 설계하여 원래 브라우저 모듈과 가능한 한 독립적으로 동작하도록 작성하였다. 추가되는 모듈들은 UI, 모델부, 통신부의 3가지로 나눌 수 있다. 모델부 모듈은 인스턴스들, 바인딩 정보와 submission들을 담당한다. UI부 모듈은 repeat, 카드 뷰들과 XForms 컨트롤들을 처리한다. 통신부 모듈은 3개의 쓰레드들을 가지는 쓰레드 풀로 이루어져있다.

4.1 모델 인스턴스 및 submission 처리부

XForms의 모델들은 인스턴스들과 submission들을 포함한다. 모델부 모듈의 중요한 역할 중 하나는 인스턴스들에 관련된 파싱과 액션 처리들이다. 우리들은 인스턴스 트리들을 파싱하고 저장하기 위하여 kDom 라이브러리[13]를 사용했다. 또한 모델은 <xforms:insert>, <xforms:delete>, <xforms:setvalue>와 같은 XForms 액션들을 위한 액션 처리부를 가진다. <xforms:setfocus>와 <xforms:refresh>는 UI부에서 처리하는 다른 액션들이다.

한편 submission 클래스는 오픈 API 요청 메시지를 생성하고 서버로부터 리턴된 메시지를 처리한다. XForms의 <xforms:submission>요소는 요청 메시지를 생성하기 위해 id, url, method, ref 등의 속성들을 가진다.

메시지를 요청할 때, 통신 쓰레드들은 UI에 의해 활성화 되고, 해당 submission 객체로부터 요청 메시지를 받아 서버에 전달한다. 쓰레드 풀은 동시에 여러 개의 비동기 호출을 위해 3개의 쓰레드를 가진다.

쓰레드가 응답을 수신하면 submission의 콜백함수를 호출한다. 콜백함수는 사용자가 XForms의 <xforms:submission>요소의 속성과 자식 액션 요소에 기술된 방법으로 응답 메시지를 처리한다. 액션들은 서버로부터 결과를 받고 난 후에 실행되는데, 보통 인스턴스 트리들의 조작을 순서대로 기술한다.

4.2 repeat 처리 모듈

repeat는 인스턴스 데이터의 반복부를 사용자 인터페이스에서 활용하기 위한 XForms 요소이다. 예를 들어 한줄 블로그의 경우 인스턴스 데이터에 여러 게시글(Post)이 들어있을 수 있는데 이러한 반복부를 HTML 페이지 상에서 표나 리스트 형태로 반복할 수 있게 해 준다. <xforms:repeat>는 ref와 id속성을 가지고 자식 트리에 임의의 요소들이 나올 수 있다.

본 구현 시스템에서는 repeat의 처리를 템플릿 처리부와 인덱스 처리부로 구분하였다. 템플릿 처리부는 UI에 보여줄 repeat의 서브 트리 내용을 계산한다. 그리고 인덱스 처리부는 *post[index['repeat-id']]*처럼 페이지의 다른 부분에서 참조할 수 있도록 각 repeat 리스트의 인덱스를 유지한다. 예를 들면 (그림 3)에서 나타난 것과 같이 상세 보기 화면에서 블로그 메인 화면에서 선택된 게시글의 인덱스를 사용할 수 있다.

4.3 카드 뷰 관리자

우리들은 XForms 페이지를 몇 개의 디스플레이 화면으

로 나누기 위한 수단으로 "Card"의 개념을 제안했다. 카드는 XForms 페이지에서 사용자가 <xforms:group> 요소에 지정한 id에 의해 정의된다. 렌더러는 <xforms:group>에 따라 별도의 카드 객체를 생성하여 포함된 블록, 인라인, 폼컨트롤 요소들의 범위를 표시한다

또한 사용자는 <xforms:submission>또는 <xforms:trigger>요소들에서 <xforms:setfocus> 액션을 통하여 카드들 사이의 이동을 제어할 수 있다. UI 모듈은 적절하게 현재 카드에 <xforms:setfocus> 액션에서 주어지는 <xforms:group>의 id를 설정하여 해당 카드를 뷰에 표시한다.

5. 결론

우리들은 자바스크립트 없이 모바일 장치들 위에 오픈 API XForms 페이지들을 기술하는 것이 가능하고 서버와 상호작용하는데 충분히 유연하다는 것을 나타내었다. 또한 오픈 API 매쉬업들을 기술하기 위한 템플릿 형식을 XForms 페이지에 제안했다. 그리고 높은 표현력과 성능을 가지는 XForms페이지를 지원하는 모바일 브라우저를 개발했다. 개발된 모바일 브라우저는 서버로 비동기 HTTP 통신을 가능하게 하고 응답결과를 처리하기 위한 콜백들을 제공한다. 마지막으로 XForms페이지가 오픈 API서비스들에 접속하기 위해 사용되는 것이 충분히 유연하고 효율적이라는 것을 한줄 블로그 시스템을 통해 증명했다. 앞으로 우리는 XForms언어로 유비쿼터스 데이터 서비스를 확장하려는 계획을 가진다.

참고문헌

- [1] J.Barton, T.Kindberg, et.al., "Sensor-enhanced mobile web clients: and XForms approach," WWW conference, pp.80-89, 2003.
- [2] R. Cardon, et.al, "Using XForms to simplify Web programming," WWW conference, pp.215-224, 2005.
- [3] M. Honkala, et.al, "Connecting XForms to databases," DMDWA Wrokshop, 2007.
- [4] Mikko Honkala and Petri Vuorimaa "A Configurable XForms implementation," ISMSE 2004.
- [5] M. Honkala, M. Pojia, "Multimodal interaction with XForms," ICWE 2006, pp.201-208, 2006. (**)
- [6] Mikko Pohja, Mikko Honkala, "Web user interaction - comparison of declarative approaches," WEBIST 2006, pp.295-302, 2006.
- [7] Raman, T. V. XForms, XML Powered Web Forms. Addison-Wesley, 2004.
- [8] Kisub Song and Kyong-Ho Lee "An Automated Generation of XForms Interfaces for Web Services," ICWS 2007.
- [9] Ruby on Rails, <http://www.rubyonrails.org/>
- [10] Web Run Time, www.s60.com/webruntime
- [11] picofoms, www.picofoms.com
- [12] World-Wide Web Consortium standards including XForms, XML Schema, XPath and Cascading Style Sheets.<http://www.w3.org>.
- [13] kxml, <http://kxml.objectweb.org/>