

장치 적응력 있는 유저 인터페이스 생성 시스템의 설계 및 구현

김영태*, 김치수*

*공주대학교 컴퓨터공학과

e-mail : zerot@kongju.ac.kr

Design and Implementation of Adaptable User Interfaces Generation System for Devices

Young-Tae Kim*, Chi-Su Kim*

*Dept of Computer Engineering, Kong-Ju National University

요 약

웹 기반 정보시스템들은 무선 장치를 사용하여 접근하고자 하는 사용자들의 요구에 따라 다양한 클라이언트 장치와 사용자, 사용자 작업에 대한 시스템 유저 인터페이스의 적합성이 요구된다.

본 논문에서는 개발자가 고 수준 마크 업 언어를 사용하여 다양한 장치와 사용자, 사용자 작업에 적합한 웹 기반 인터페이스를 명세할 수 있도록 하는 접근방법을 제시하고자 한다. 본 논문에서 제시하는 접근방법을 통한 단일 인터페이스 명세는 실행 시에 다양한 웹 장치, 사용자, 작업에 대하여 적합한 인터페이스를 자동으로 제공함으로써 시스템 개발에 유연성과 일관성을 부여할 수 있게 된다.

1. 서론

21세기 초의 매우 중요한 기술적 발전중의 하나는 모바일 폰과 같은 장치에 의해 제공 되어지는 신속성과 인터넷에서 이용될 수 있는 엄청난 양의 정보를 동반한다는 것이다.

많은 정보시스템은 데스크 탑 애플리케이션, 웹 기반 인터페이스, 모바일 폰, PDA 등을 포함하고 있으며, 점점 더 많은 사람들이 셀룰러 폰이나 PDA와 같은 무선 장치들을 구입하고 있으며 언제 어디서나 온라인에 접속하기를 원한다[nokia developer].

따라서 많은 웹 기반 정보시스템은 다양한 사용자와 사용자 업무, 디스플레이 장치, 네트워크에 대한 적합한 유저 인터페이스의 제공을 요구하고 있으며, 이것은 개발자들이 원해왔던 통일된 적합성의 제공을 의미한다[1].

다양한 화면 크기를 갖는 상이한 장치들을 통해 서비스를 제공하려면 시스템은 사용자의 장치, 사용자의 선택에 자동적으로 적응되어야 하며, 요구되는 콘텐츠 형식에 맞게 분할 및 변환이 필수적으로 이루어져야 한다[2].

또한 상이한 사용자와 사용자 작업에 적용할 수

있는 유저 인터페이스를 필요로 한다. 그러나 콘텐츠의 서비스가 자주 변경되고, 모든 서비스들을 최신으로 유지시키고자 할 때 지속성이 요구되어지고 시간을 많이 소비하게 된다. 결과적으로 많은 수의 인터페이스가 개발되고 유지되어야 한다.

본 논문에서는, 소프트웨어 개발자가 다양한 사용자와 장치에 대한 인터페이스를 좀 더 쉽게 설계하고, 구축하며 전개하도록 하기위해 특정한 장치 특징들, 콘텐츠의 표현 형식뿐만 아니라 사용자와 특정한 작업에 대한 적응력 있는 유저 인터페이스를 제공할 수 있는 개발방법을 제시하고자 한다.

2. 관련연구

2.1 컴포넌트 기반 개발방법

컴포넌트는 동일한 인터페이스를 가지며 동일한 사용사례를 갖는 다른 컴포넌트와 대체될 수 있다. 이러한 특징으로 인해서 시스템은 보다 유연해질 수 있으며 개발 과정에서 반복적 개발에 도움을 줄 수 있고 소프트웨어 시스템의 유지보수 비용을 절감할 수 있다[3].

컴포넌트 기반 유저 인터페이스는 대개 다양한 사

용자와 사용자 작업의 적합성을 지원한다[4]. 그러나 클라이언트 장치에서 실행되고, 서버에서는 실행되지 않는 thick-client 인터페이스를 제공하고 장치 적응 능력은 거의 제공하지 않는다.

2.2 자동 변환기

다양한 단일 자동 변환기들은 HTML을 WML로의 변환을 지원한다[5]. 웹 개발자들은 무선 장치를 지원할 환경을 만들기 위해 자동적으로 변환을 수행하는 소프트웨어를 사용한다.

자동 변환 기술들이 WML 브라우저를 위해 HTML을 WML로의 변환을 적당히 성공시키지만, 대부분 어려운 코드화 작업으로 되어있기 때문에 웹 서비스의 다양한 상황에 맞는 최적의 사용자 상호작용을 제공하지 못하며, 사용자와 사용자의 작업에 대한 적응을 지원하지는 않는다.

2.3 XML/XSL 변환

적응력 있는 접근방법을 구현하기 선택할 수 있는 하나의 선택이며 특정한 장치에 대한 출력을 생성하는 대신 XML 형식화된 데이터를 생성하기 위해 하나의 페이지 집합을 작성할 수 있다.

시스템은 XML 서술 인터페이스 콘텐츠를 갖고, 그것을 HTML, WML, 요구하는 장치에 의존적인 다른 XML 형식으로 변환한다[6].

이 접근방법을 사용해서 HTML 페이지를 생성하기 위해서 XSL 스타일시트와 스타일시트를 적용하기 위한 방법이 필요하다.

2.4 사용자 태그와 태그 라이브러리

개발자들은 웹 기반 애플리케이션을 개발하기 위해 JSP를 사용한다. JSP의 가장 큰 장점은 개발자들로 하여금 사용자 태그라 불리는 태그 집합을 생성할 수 있게 해준다는 것이다. 사용자 태그는 하나의 태그 라이브러리 정의로 코드를 재사용할 수 있다. 본 논문에서는 JSP의 사용자 태그를 다양한 장치를 위한 인터페이스를 생성할 수 있는 태그 집합을 작성하는데 사용한다.

3. AUIDD의 설계

적응력 있는 유저 인터페이스는 Adaptable User Interface for Diverse Devices(AUIDD) 페이지라 불리는 AUIDD 요소들의 집합을 사용하여 생성하게 된다.

본 논문에서의 적응력 있는 접근방법은 개발자가 어느 장치에나 만족하는 유저 인터페이스를 생성할 수 있는 단일 시스템만을 생성하도록 하여 개발시간 단축과 작업부하 감소를 제공하고자 한다.

3.1 적응력 있는 유저 인터페이스의 요구사항

다음은 본 논문에서 제시하는 적응력 있는 유저 인터페이스를 위한 요구사항이다.

1) 동적 마크업 언어 생성

다양한 장치들이 다른 마크업 언어를 지원하기 때문에 시스템은 사용자가 애플리케이션을 실행하기 위해 사용할 것으로 예상되는 장치에 적절한 마크업 언어를 동적으로 생성할 수 있는 능력을 제공해야 한다. 클라이언트 장치가 서비스를 요청할 때 시스템은 장치를 인지하고 마크업 언어를 결정하며 레이아웃을 표현하게 된다.

2) 다양한 장치에 대한 적응

모바일 폰이 낮은 해상도의 작은 화면, 한정된 메모리와 처리능력, 데스크 탑 시스템보다 적은 대역폭을 갖는다는 것을 기억해야 한다. 또한 많은 장치들은 정보를 디스플레이 하는 스크린 크기가 다르다. 실행 시에 자동적으로 주어진 장치의 화면 크기나 메모리 능력에 따라 요소들이 디스플레이 될 수 있어야 한다.

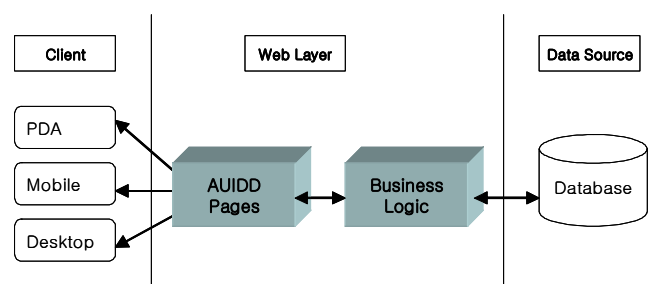
3) 사용자와 사용자 환경에 대한 적응

사용자의 활동 패턴을 감시하고, 시스템에 의해 제공되는 인터페이스나 콘텐츠를 사용자 선택에 따라 적응되도록 자동적으로 조절되어야 한다.

3.2 전체 시스템 구조

AUIDD 시스템은 4-tier 구조를 갖는다. 클라이언트는 데스크 탑 PC, WML 기반 브라우저를 사용하는 무선장치나 PDA가 될 수 있다.

다음 [그림 1]은 AUIDD를 사용하는 시스템의 전체 구성도이다.



[그림 1] AUIDD 시스템의 전체 구성도

AUIDD 페이지는 클라이언트와 비즈니스 로직 사이에 위치한다. 웹 레이어는 디스플레이 되는 데이터와 포맷을 검색한다. AUIDD 표현 페이지에서 비즈니스 로직을 분리하는 것은 애플리케이션의 설계에 유연성을 부여하기 위함이다. 다중 사용자 인터페이스는 비즈니스 로직에서 명확하게 정의된 인터페이스의 어떤 변화 없이 표현 계층으로 구성되고 조립

된다.

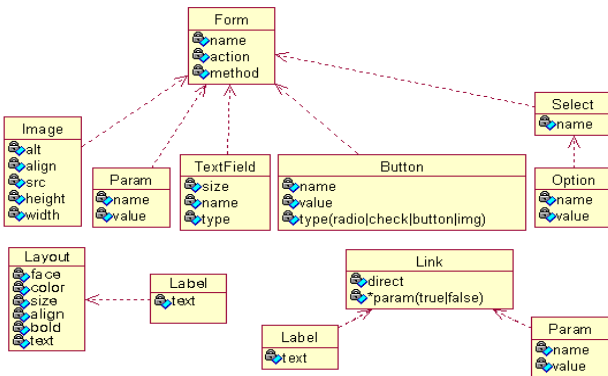
3.3 AUIDD 요소의 설계

AUIDD 페이지는 AUIDD 요소로 이루어진다. 따라서 상이한 언어로부터 공통 요소들을 추출하여 고수준 요소들의 목록을 생성해야 한다.

AUIDD 페이지의 기본 구조를 명세하기 위해 "Template"이라 불리는 요소를 사용하게 된다. 문서 구조 요소는 HTML 문서와 WML 문서에서 모두 필요한 주된 구조를 제공해야 한다.

1) 일반적인 UI 요소들

각각의 인터페이스는 UI 요소들의 집합으로 이루어진다. 다음 [그림 2]는 해당되는 장치를 위한 페이지에 콘텐츠를 표현할 수 있는 적절한 마크업 언어를 생성하는데 사용하게 될 UI 요소들의 집합을 보여주는 다이어그램이다.



[그림 2] 일반적인 UI 요소들

2) UI 관리 요소들

설계자는 각 장치에 해당되는 각각의 페이지에 대해서 같은 양의 작업을 하기 원할 것이다. UI의 상이한 유형을 관리하기 위해서 사용자, 작업, 장치의 3요소가 사용된다. 이 요소들이 동작하는 방법은 사용자의 유형이나 장치의 유형 등에 따라 AUIDD 페이지 레이아웃의 일부분을 숨기는 것이다.

3) 화면 분할 요소들

어떤 환경에서 다른 환경으로 유저 인터페이스를 적절하게 변경할 수 있어야 한다. 따라서 필요한 경우 다중 유저 인터페이스가 동시에 생성될 수 있어야 한다.

모든 아이템이 하나의 디스플레이 장치에 분명하게 디스플레이 되지 못할 때 화면을 다중 화면으로 분할하는 내부적인 변환을 갖게 하기 위한 Group 요소들 group/grouptr/grouptd를 설계했다. Group 요소들은 화면의 UI 요소들의 디스플레이를 관리하고, 콘텐츠가 다른 화면이나 card로 이동되어야 하

는지를 결정한다.

4. AUIDD의 구현

1) 템플릿 태그의 구현

템플릿 태그는 AUIDD 페이지 상단에 나타나며 장치의 식별, AUIDD 페이지 구조의 제공, 닫힌 태그 </AUIDD:template>로 레이아웃의 출력을 수행하는 매우 중요한 태그이다. [그림 3]는 템플릿 태그 클래스의 일부로서 요구하는 장치에 대해 각각 적합하게 미리 보호된 WML이나 HTML 문서 구조를 생성하는 코드이다.

```

if(device="html")
    output.append("<html><head><title>"+ name+ "</title></head><body>")
else{
    output.append("<?xml version='1.0'?>");
    pageContent.getResponse().setContentType("text/vnd.wap.wml");
    output.append("<!DOCTYPE wml PUBLIC '//WAPFORUM/DTD WML 1.1//EN'
        http://www.wapforum.org/DTD/wml_1.1.xml"><wml>");
}
    
```

[그림 3] 문서 구조를 생성하는 코드의 일부 애플리케이션을 디스플레이 하기 위해서 사용될 장치 정보, 사용자 정보, 사용자 선택 사항 등을 데이터베이스에 입력하게 되며, 입력된 정보를 기준으로 하여 적절한 유저 인터페이스를 생성하게 된다.

2) 페이지 흐름 관리 요소의 구현

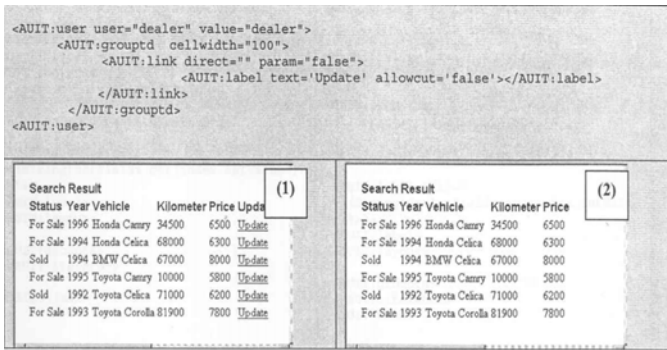
<AUIDD:group>, <AUIDD:grouptr> 그리고 <AUIDD:grouptd> 태그는 group 셀에 포함된 AUIDD 요소들의 자동 레이아웃 설정에 사용되며, 여러 가지 적응 작업을 완성하게 된다. 다음 [그림 4]는 다양한 장치를 위해 분할된 화면 사이의 링크를 보여주는 예이다.

For devices support HTML	For devices support WML
<pre> <html> <body> Right (link to 10) Left (link to 00) Screen 00 Screen 01 Down (link to 01) Down (link to 11) Up (link to 00) Up (link to 01) Screen 10 Screen 11 Right (link=11) Left (link to 10) </body> </html> </pre>	<pre> <WML> <Card id="00"> R(link to 10) </Card> <Card id="01"> L(link to 00) </Card> <Card id="10"> D(link to 10) U(link to 00) </Card> <Card id="11"> D(link to 11) U(link to 01) </Card> <Card id="10"> R(link to 11) </Card> <Card id="11"> L(link to 10) </Card> </WML> </pre>

[그림 4] 다양한 장치를 위한 링크를 보여주는 예

3) UI 관리 요소의 구현

UI 관리 요소들은 UI 요소들의 레이아웃을 관리하기 위한 요소들의 집합으로 UI 요소들을 사용자, 장치, 작업에 따라 적합하도록 디스플레이 한다. 다음 [그림 5]은 사용자에게 따라 다른 화면을 보이도록 하기 위해 <AUIDD:user> 태그를 사용한 예이다.



[그림 5] 사용자 적응에 대한 코드 예

4) UI 요소들의 구현

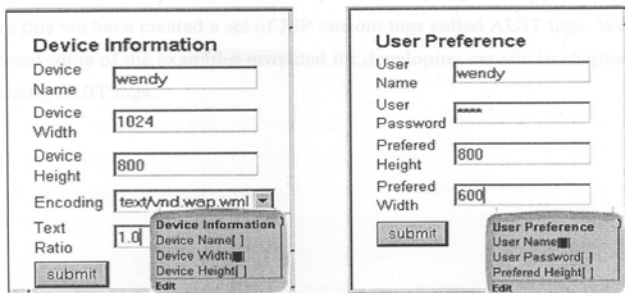
<AUIT:label>과 <AUIT:layout> 태그는 꽤 간단한 태그들로 <AUIT:label>은 장치에 일반적인 텍스트를 디스플레이 하고, <AUIT:layout>은 크기나 색상을 설정하는 것과 같이 텍스트의 형식을 지정하는데 사용된다. 다음 코드는 다양한 장치를 위한 상이한 텍스트를 디스플레이 하는 코드이다.

```
<AUIT:device name="IE" value="html">
  <AUIT:label text="Search Result"/>
</AUIT:device>
<AUIT:device name="Nokia" value="wml">
  <AUIT:label text="Result"/>
</AUIT:device>
```

[그림 6] 텍스트 디스플레이 제어를 위한 코드 예

5) 선택 기반 인터페이스 구성

개별적인 사용자의 선택에 따라 적합한 인터페이스를 제공해야한다. 추가되는 새로운 종류의 장치, 사용자 또는 작업들의 정보는 개발 시에는 알 수 없기 때문에 사용자가 인터페이스에 접근하고 수정할 수 있도록 프로그래밍된 인터페이스를 제공한다. 사용자가 사이트에 접근할 때마다 관련 장치 정보는 데이터베이스 내의 사용자 데이터와 비교해서 사용된다. 다음 [그림 7]은 사용자가 선택 정보에 관한 인터페이스를 명세할 수 있도록 하는 화면 예이다.



[그림 7] 시스템 데이터 입력을 위한 인터페이스 예

5. 결론 및 향후 연구과제

본 논문에서는 다양한 장치, 사용자, 사용자 작업에 따라 적합한 유저 인터페이스를 실행 시에 동적으로 생성하여 제공할 수 있는 방법을 제시하였다.

본 논문에서 제시한 접근방법을 통한 인터페이스의 구현은 HTML과 유사한 JSP 사용자 태그를 사용하므로 사용이 용이한 고수준의 마크 업 언어를 제공하게 되며, 다양한 장치, 사용자, 사용자 작업에 강력한 적응력을 갖는 쉬운 구조를 제공하게 된다.

향후 연구과제로는 본 논문에서 사용된 사용자 태그를 확장하여 보다 많은 기능을 지원할 수 있도록 하여야 하며, 본 논문에서 예를 든 HTML과 WML 이외의 다른 마크 업 언어들을 지원할 수 있도록 시스템을 확장할 필요가 있다.

참고문헌

- [1] Van der Donckt,J., Limbourg,Q., Florins,M, Oger,F., Macq,B, "Synchronized, model based design of multiple user interfaces", Workshop on Multiple User Interfaces over the Internet, 2001
- [2] Jari Korva, Johan Plomp, Peter Määttä, Meija Mesto "On-line Service adaptable for mobile and fixed terminal devices", 2000
- [3] Peter Herzum, Oliver Sims, "Business Component Factory", Wiley Computer Publishing, 2000
- [4] Stephanidis, C. Concept of Unified User Interfaces, In Interfaces for All Concepts, Method and Tools, LAurence Erlbaum Associates, 2001
- [5] Fox,A., Gribble,S., Chawathe,Y., Brewer,E., "Adapting to Network and Client Variation Using Infrastructural Proxies:lessons and perspectives", IEEE Personal Communications 5(4), 1998
- [6] "Adaptable Collaboration for Wireless Platforms", IEEE Internet Computing, 2001