

다양한 장치를 위한 적응력 있는 유저 인터페이스 생성 시스템의 설계 및 구현

김치수[†] · 김영태^{**}

요 약

많은 웹 기반 정보시스템은 여러 가지 클라이언트 장치와 사용자들, 사용자 작업에 대한 시스템의 유저 인터페이스에 관한 적응을 요구한다. 본 논문에서는 개발자가 고수준 마크업 언어를 사용하여 웹 기반 인터페이스를 명세할 수 있는 웹 기반 정보시스템에 적합한 클라이언트 인터페이스를 제공하기 위한 새로운 접근방법을 제시하고자 한다. 단일 인터페이스 명세는 실행 시에 다중 웹 장치에 대한 인터페이스를 자동으로 제공하기 위한 것이다.

본 접근방법은 개발자들이 웹 기반 유저 인터페이스를 현재 다른 접근 방법들보다 더 쉽게 구성하고 유지할 수 있게 할 것이다.

키워드 : 적응, 유저 인터페이스, 다중 웹 장치

Design and Implementation of Adaptable User Interfaces Generation System for Diverse Devices

Kim Chi Su[†] · Kim Young Tae^{**}

ABSTRACT

Many web-based information systems require degrees of adaptation of the system's user interfaces to different client devices, users and tasks.

We describe a new approach to providing adaptable client interfaces for web-based information systems that allow a developer to specify a web-based interface using a high-level mark-up language. At run-time this single interface description is used to automatically provide for multiple web devices.

This approach allows developers to much more easily construct and maintain web-based user interfaces than other current approaches.

Key Words : Adaptation, User Interface, Multiple Web Devices

1. 서 론

21세기 초의 매우 중요한 기술적 발전 중의 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[1].

따라서 많은 웹 기반 정보시스템은 다양한 사용자와 사용자 업무, 디스플레이 장치, 네트워크에 대한 적합한 유저 인

터페이스의 제공을 요구하고 있으며, 이것은 개발자들이 원해왔던 통일된 적합성의 제공을 의미한다[2].

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[3].

또한 상이한 사용자와 사용자 작업에 가장 잘 맞는 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 대해 가장 잘 맞는

[†] 종신회원 : 공주대학교 컴퓨터공학부 교수(공학연구원)
^{**} 준 회원 : 공주대학교 컴퓨터공학과 박사과정
논문접수 : 2005년 9월 14일, 심사완료 : 2005년 10월 17일

유저 인터페이스를 제공할 수 있는 개발방법을 제시하고자 한다.

2. 관련 연구

2.1 컴포넌트 기반 개발방법

컴포넌트 기반 유저 인터페이스는 대개 다양한 사용자와 사용자 작업의 적합성을 지원한다[4,5]. 그러나 클라이언트 장치에서 실행되고, 서버에서는 실행되지 않는 thick-client 인터페이스를 제공하고 장치 적응 능력은 거의 제공하지 않는다.

2.2 자동 변환기

자동 변환 기술들이 WML 브라우저를 위해 HTML을 WML로의 변환을 상당히 성공시키지만, 대부분 어려운 코드화 작업으로 되어있기 때문에 웹 서비스의 다양한 상황에 맞는 최적의 사용자 상호 작용을 제공하지 못하며 사용자와 사용자의 작업에 대한 적응을 지원하지는 않는다[6].

2.3 Palm의 웹 클리핑

Palm의 웹 클리핑 접근 방법은 HTML 문서의 많은 내용을 간소화된 WML 버전으로 만들기 위해 자르는 것이다. 전형적으로 웹 클리핑은 사용자 질의에 응답하는 CGI 스크립트에 의해서 작고 동적인 웹 페이지를 생성한다[7].

클리핑의 목적은 디스플레이 요구사항과 대역폭 사용을 최소화하는 것이며 웹 페이지 전송을 위한 것이 아니라 클리핑이 목적이다.

웹 클리핑은 사용자가 그래픽의 사용을 최소화해야 하며 WML 장치를 지원하지 않는다는 제약을 갖는다.

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.

WAP/WML taglib는 WAP/WML 개발자들을 위해 개발된 JSP 태그 라이브러리로 두 개의 메인 태그 즉, "WmlOn"과 "WmlOff"가 있다. 사용자로 하여금 일반적인 브라우저와 WAP 브라우저에 대한 여러 가지 코드를 하나의 JSP 페이지로 합치도록 한다[8].

Codejava Escape Taglib는 WAP/WML 개발자들이 HTML을 WML로 변환하는데 편리하게 사용된다. 이 태그 라이브러리는 두 개의 body 태그 즉, "Escape"와 "WmlEscape"를 갖는다. Escape 태그는 HTML 태그를 WAP 브라우저에서 볼 수 있도록 변환해주며, WmlEscape 태그는 비 WAP 브라우저에서 HTML 코드를 유지하면서 적합한 WML 코드로 변환한다. WML 파일은 제한된 사이즈를 갖기 때문에 이 태그는 콘텐츠를 나누어 적합한 JSP 파일을

생성할 수 있다[9].

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

3.1.1 동적 마크 업 언어 생성

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

3.1.2 다양한 장치에 대한 적응

오늘날의 데스크 탑 장치에서 잠재적으로 무한한 용량의 램과 하드디스크 영역이 사용가능함에 따라, 개발자들은 소프트웨어 설계에 보다 적은 제약을 받는다. 그러나 모바일 폰은 낮은 해상도의 작은 화면, 한정된 메모리와 처리능력, 데스크 탑 시스템보다 적은 대역폭을 갖는다. 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.

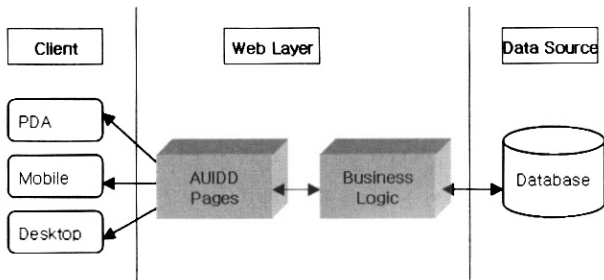
본 논문에서는 적응력 있는 접근방법을 통하여 실행 시에 자동으로 각각의 특별한 장치들을 위해 알맞은 콘텐츠를 생성할 수 있도록 한다.

3.1.3 사용자와 사용자 환경에 대한 적응

많은 유저 인터페이스나 요소들은 어떤 사용자에게는 적합하지만 특별한 사용자 규칙이나 부속 작업이 이행되기에는 적합하지 않은 경우도 있다. 대부분의 적응력 있는 시스템들은 사용자와 사용자 선택에 대한 적응 능력에 초점을 맞춘다. 개별 사용자의 요구사항과 선택은 소프트웨어 시스템을 사용하면서 바뀔 것이다. 적응력 있는 시스템은 사용자의 활동 패턴을 감시하고, 시스템에 의해 제공되는 인터페이스나 콘텐츠를 사용자 선택에 따라 적응되도록 자동적으로 조절한다.

3.2 전체 시스템 구조

AUIDD 시스템은 4-계층 구조를 갖는다. 클라이언트는



(그림 1) AUIDD 시스템의 전체 구성도

데스크 탑 PC, WML 기반 브라우저를 사용하는 무선장치나 PDA가 될 수 있다. (그림 1)은 시스템의 전체 구성도이다.

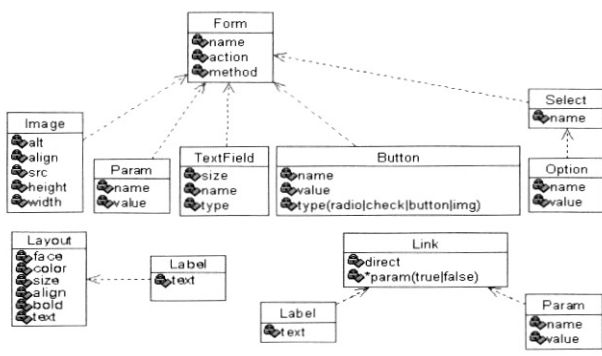
AUIDD 페이지는 클라이언트와 비즈니스 로직 사이에 위치한다. 웹 레이어는 디스플레이 되는 데이터와 포맷을 검색한다. AUIDD 표현 페이지에서 비즈니스 로직을 분리하는 것은 애플리케이션의 설계에 유연성을 부여하기 위함이다. 다중 사용자 인터페이스는 비즈니스 로직에서 명확하게 정의된 인터페이스의 어떤 변화 없이 표현 계층으로 구성되고 조립된다.

3.3 AUIDD 요소의 설계

3.3.1 일반적인 UI 요소들

각각의 인터페이스는 UI 요소들의 집합으로 이루어진다. AUIDD 페이지에서 명세한 유저 인터페이스를 만들 수 있는 UI 요소들의 집합을 설계해야 한다. 우선 해당되는 장치를 위한 페이지에 콘텐츠를 표현할 수 있는 적절한 마크 업 언어를 생성하는데 사용하게 될 UI 요소들의 목록을 생성해야 한다. HTML과 WML의 표준 DTD로부터 몇 개의 공통적인 요소를 추출할 수 있다.

(그림 2)는 해당되는 장치를 위한 페이지에 콘텐츠를 표현할 수 있는 적절한 마크 업 언어를 생성하는데 사용하게 될 UI 요소들의 집합을 보여주는 다이어그램이다. 사용자 입력 요소는 장치를 지원하는 마크업 언어에 상관없이 사용자 입력을 받는 메커니즘을 제공한다. 요소들과 요소의 속성들은 HTML에서 사용되는 것과 유사한 이름을 갖는 것을



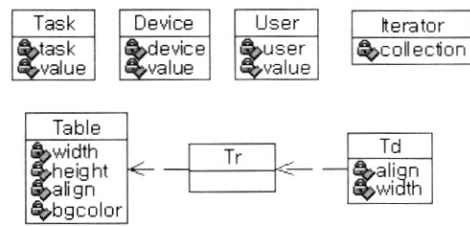
(그림 2) 일반적인 UI 요소들

볼 수 있다. 개발자들의 쉬운 선택을 위해 HTML과 같은 방법을 따르도록 설계했다.

3.3.2 UI 관리 요소들

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.

(그림 3)의 UI 관리 요소는 UI 요소들의 관리를 책임진다. UI 관리 요소가 UI 요소들이 디스플레이 될지, 어떻게 디스플레이 될지를 결정한다. (그림 3)에서 Task, Device, User 요소가 유사한 구조와 응용을 갖고 있지만, 서로 다른 조건들을 통해 레이아웃의 보임과 숨김을 관리한다. 예컨대 Task 요소는 시작 태그와 종료 태그 사이의 콘텐츠를 사용자가 실행할 수 있는 작업이면 보여주고, 그렇지 않으면 감추는 관리를 한다.



(그림 3) UI 관리 요소들

3.3.3 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 하는지를 결정한다.

<표 1>의 “content” 위치에 모든 UI 요소들이 적합해야 한다. 여러 화면에 출력하기 위해 분할된 콘텐츠와 같이 “group/grouptr/grouptd” 요소들 없이 어떤 알고리즘 기능을 수행하기 위해 “group” 요소의 속성으로 “width”나 “height” 값을 명시할 수 있다.

장치의 한 화면에 디스플레이 되기에 충분하지 않아서 다른 화면에 있는 다른 요소들의 그룹을 링크로 디스플레이하기 위한 페이지 레이아웃을 관리하기 위해 나눌 수 있는 셀을 명시하고, 페이지의 콘텐츠를 나누기 위한 알고리즘을 사용하여 몇 개의 화면으로 출력한다. 나눌 수 있는 셀을 명시하는 목적은 알고리즘의 응답 후에 장치의 화면 크기에 따라 최소로 나눌 수 있는 셀 몇 개 혹은 전체가 하나의 화면에 그룹화 되어 나타난다.

〈표 1〉 "group" 요소들의 사용

On HTML, the scrip is like	On Adaptable Approach
<pre><table> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> </tr> </table></pre>	<pre><group> <grouptr> <grouptd>content</grouptd> <grouptd>content</grouptd> </grouptr> <grouptr> <grouptd>content</grouptd> <grouptd>content</grouptd> </grouptr> </group></pre>

4. AUIDD의 구현

4.1 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.

AUIDD를 사용하면 모든 장치에 대해 하나의 AUIDD 페이지가 존재하므로 AUIDD 페이지에서 각각의 장치 정보를 보호해야 한다. 각각의 AUIDD 페이지 처음에 장치 정보를 보호해야 하고, 다른 AUIDD 태그에 의해 사용될 수 있는 페이지 변수의 데이터를 저장해야 한다. 콘텐츠는 AUIDD 페이지에 의해 요구하는 장치가 지원하는 형식에 맞게 동적으로 생성된다. 헤더 정보로부터 얻은 사용자 유형에 따라 장치 정보의 저장을 위한 페이지 변수를 갖게 되며, 이것은 다른 AUIDD 태그에 의해 사용될 수 있도록 "templateTag" 클래스에 놓아야 한다.

AUIDD 페이지의 대부분의 사용자 태그는 장치 정보를 필요로 한다. 장치 유형의 명세 후에 장치 정보는 setAttribute() 메소드를 사용하여 속성으로 저장될 수 있다. 이것은 getAttribute() 메소드를 사용하는 모든 요소들에 의해 접근 가능하고 검색 가능하다.

4.2 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.

분할된 화면 정보는 네 개의 StringBuffer"로 저장되고 첫 번째 "StringBuffer"만 출력된다. 다른 세 개의 페이지를 보기 위해 사용자는 링크를 클릭하게 된다.

WML에서 "table" 태그는 HTML과는 다르다. WML은 셀에서 텍스트와 이미지 형식 요소만을 허용한다. WML을 사용해서 (그림 5)의 화면1에서와 같은 테이블 레이아웃을 얻는 것은 HTML에서처럼 쉽지 않다. 하나의 방법은 각각의 셀에 대한 정렬을 수동으로 명시하고, 중요하지 않은 문자들을 삭제하는 것이다. 이 접근방법을 구현하려면 각각의 셀에 디스플레이 되어야 하는 문자의 수를 세어야 한다. 예를 들어 한 줄을 디스플레이하기 위한 셀의 폭을 100픽셀로 할당할 때, 데이터베이스에 한 문자가 각각의 장치에서 차지하는 픽셀 수 같은 데이터를 미리 저장해야 한다. PDA에서 한 문자가 5픽셀을 차지한다면 라인마다 디스플레이 가능한 문자의 수는 100/5=20이다. 이 예는 (그림 5)의 화면2에서 보여준다.

위의 값으로 각각의 셀에 포함될 수 있는 문자의 최대 수를 계산할 수 있다. 각각의 셀마다 나머지 문자들은 없애고 최초의 5 문자를 디스플레이 한다.

4.3 UI 관리 요소의 구현

UI 관리 요소들은 UI 요소들의 레이아웃을 관리하기 위한 요소들의 집합으로 UI 요소들을 사용자, 장치, 작업에 따라 적합하도록 디스플레이 한다. 이 작업을 수행하기 위해 세 가지 요소 <AUIDD : user>, <AUIDD : task>, <AUIDD : device>를 사용한다.

이 UI 제어 태그를 사용하려면 사용자, 작업, 장치의 고유 값을 페이지의 매개 변수나 속성으로 취해야 한다. 장치

HTML을 지원하는 장치	WML을 지원하는 장치								
<pre><html> <body> <table border="1"> <tr> <td>Right (link to 01) 화면 00 Down (link to 10)</td> <td>Left (link to 00) 화면 01 Down (link to 11)</td> </tr> <tr> <td>Up (link to 00) 화면 10 Right (link to 11)</td> <td>Up (link to 01) 화면 11 Left(link to 10)</td> </tr> </table> </body> </html></pre>	Right (link to 01) 화면 00 Down (link to 10)	Left (link to 00) 화면 01 Down (link to 11)	Up (link to 00) 화면 10 Right (link to 11)	Up (link to 01) 화면 11 Left(link to 10)	<pre><WML> <table border="1"> <tr> <td> <Card id="00"> R (link to 01) Card 00 D(link to 10) </Card> </td> <td> <Card id="01"> L (link to 00) Card 01 D(link to 11) </Card> </td> </tr> <tr> <td> <Card id="10"> U (link to 00) Card 10 R (link to 11) </Card> </td> <td> <Card id="11"> U (link to 01) Card 11 L(link to 10) </Card> </td> </tr> </table> </WML></pre>	<Card id="00"> R (link to 01) Card 00 D(link to 10) </Card>	<Card id="01"> L (link to 00) Card 01 D(link to 11) </Card>	<Card id="10"> U (link to 00) Card 10 R (link to 11) </Card>	<Card id="11"> U (link to 01) Card 11 L(link to 10) </Card>
Right (link to 01) 화면 00 Down (link to 10)	Left (link to 00) 화면 01 Down (link to 11)								
Up (link to 00) 화면 10 Right (link to 11)	Up (link to 01) 화면 11 Left(link to 10)								
<Card id="00"> R (link to 01) Card 00 D(link to 10) </Card>	<Card id="01"> L (link to 00) Card 01 D(link to 11) </Card>								
<Card id="10"> U (link to 00) Card 10 R (link to 11) </Card>	<Card id="11"> U (link to 01) Card 11 L(link to 10) </Card>								

(그림 4) 다양한 장치를 위한 링크를 보여주는 예

메이커	모델	시리즈	연식	가격	색상	미션	주행거리
현대	아반테XD	1.5 DELUXE	2003/01	919 만원	흰색	오토	37,254 km
현대	뉴 베르나	1.5 GV	2005/01	857 만원	흰색	오토	7,990 km
삼성	SM3	FE	2004/03	1,060 만원	진주	오토	30,000 km
기아	스펙트라	1.5 GOLD	2004/03	850 만원	은색	오토	가속 색상 미션 주행 919 흰색 오토 37254 857 흰색 오토 7990 1060 진주 오토 30000 850 은색 오토 30000

메이커	모델	시리즈	연식	가격	색상	미션	주행거리
현대	아반테XD	1.5 SPORTS	2003/02	890 만원	검정	수동	70,000 km
현대	아반테XD	2.0 GOLD	2003/11	1,110 만원	은색	오토	25,000 km
현대	아반테XD	1.5 DELUXE	2003/06	1,010 만원	흰색	오토	26,000 km
대우	라세티	MAX	2003/06	950 만원	은색	오토	980 검정 수동 70000 1110 은색 오토 25000
기아	세라토	1.5 GOLD	2004/06	1,050 만원	빨강	오토	1010 흰색 오토 26000 950 은색 오토 40000 1050 빨강 오토 13000

(그림 5) 두 장치에서 화면 분할이 실행되는 페이지

task="car"								task="dealer"			
검색 결과								검색 결과			
회사	모델	연식	가격	색상	미션	주행		판매자	주소	전화	e-mail
현대	아반테XD	200301	919	흰색	오토	37254		김우진	서울 강서구 등촌동 14-7 202	011-794-2052	black3077@hanmail.net
현대	뉴 베르나	200501	857	흰색	오토	7990		오경희	경기 성남시 수정구 단대동 83	016-364-9906	ohk0707@yahoo.com
삼성	SM3	200403	1060	진주	오토	30000		배승진	전남 목포시 흥해동 532~1300	016-876-3119	bsh0329@hanmail.net
기아	스펙트라	200306	850	은색	오토	30000		김철중	충북 청주시 흥덕구 복대2동 849	016-663-3591	kia2718@naver.com

(그림 6) "searchresult.jsp"

명세 정보는 <AUIDD : template> 내에 헤더 정보를 통해서 보호된다. 데이터는 JSP에 의해 제공되는 "getParameter()" 메소드를 사용하는 폼이나 링크를 통하여 얻어질 수 있다. 이 값을 받은 후에 AUIT 태그에 속성으로 명세된 값과 비교한다.

(그림 6)의 두 페이지는 같은 AUIDD 페이지인 "searchresult.jsp"로 작업에 따라 다른 화면을 보이기 위해 <AUIDD : task> 태그를 사용한 예이다. 첫 번째는 사용자가 자동차 목록을 검색한 결과이고, 두 번째 페이지는 사용자가 상인 목록을 검색한 결과로서 다른 작업 값이 적용되어 다른 레이아웃이 생성되었다.

4.4 UI 요소들의 구현

<AUIDD : label>과 <AUIDD : layout> 태그는 꽤 간단한 태그로 <AUIDD : label>은 장치에 일반적인 텍스트를 디스플레이 하고, <AUIDD : layout>은 크기나 색상을 설정하는 것과 같이 텍스트의 형식을 지정하는데 사용된다.

모바일 폰과 같은 장치들은 전체 내용을 디스플레이할 수 없기 때문에 <AUIDD : label> 요소의 텍스트를 서버에서 처리하도록 위해 넘길 수 있다. 이것은 <AUIDD : label>의 텍스트 속성과 같은 텍스트 정보를 AUIDD 페이지에 직접 기술하는 대신 텍스트 정보를 명세하기 위해 태그를 설계하

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

(그림 7) 텍스트 디스플레이 제어를 위한 코드 예

는 중요한 이유이다. (그림 7)은 다양한 장치를 위한 상이한 텍스트를 디스플레이 하는 코드이다.

모바일 폰에 테이블을 디스플레이하기 위해서는 장치에 맞도록 단어의 일부분을 잘라낼 필요가 있다. 그러나 주행거리나 자동차 가격처럼 의미의 변화 없이는 잘라낼 수 없는 단어들은 코드에 속성을 명세해야 한다. 이를 위해 "label" 태그 안에 이 기능을 명세하는 "allowcut" 속성을 사용한다. "true" 값은 필요한 경우 자를 수 있는 텍스트를 의미한다.

다음은 allowcut 속성의 사용 예이다.

```
<AUIT : layout face="Arial, Helvetica, sans-serif" size="+3" bold="b" color="#101077">
  <AUIT : label text="Search Result" allowcut="true">
</AUIT : label>
</AUIT : layout>
```

장치 정보		사용자 정보	
장치명	zerotk	사용자명	zerot
장치번호	1024	비밀번호	*****
장치높이	800	너 비	800
인코딩	text/vnd.wap.wml	높 이	600
<input type="button" value="전송"/>		<input type="button" value="전송"/>	

(그림 8) 시스템 데이터 입력을 위한 인터페이스 예

4.5 선택 기반 인터페이스 구성

개별적인 사용자의 선택에 따라 적합한 인터페이스를 제공해야 한다. 추가되는 새로운 종류의 장치, 사용자 또는 작업들의 정보는 개발 시에는 알 수 없기 때문에 사용자가 인터페이스에 접근하고 수정할 수 있도록 프로그램화된 인터페이스를 제공한다. 사용자가 사이트에 접근할 때마다 관련 장치 정보는 데이터베이스 내의 사용자 데이터와 비교해서 사용된다. (그림 8)은 사용자가 선택 정보에 관한 인터페이스를 명세할 수 있도록 하는 화면 예이다.

5. 결론 및 향후 연구과제

본 논문에서는 다양한 장치, 사용자, 사용자 작업에 따라 적합한 유지 인터페이스를 실행 시에 동적으로 생성하여 제공할 수 있는 방법을 제시하였다. 본 논문에서 제시한 접근 방법을 통한 인터페이스의 구현은 HTML과 유사한 JSP 사용자 태그를 사용하므로 사용이 용이한 고수준의 마크업 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 예들 든 HTML과 WML 이외의 다른 마크업 언어들을 지원할 수 있도록 시스템을 확장할 필요가 있다.

참 고 문 헌

[1] "Nokia WAP Toolkit Developer's Guide", version2.1, "Using the Palm OS Emulator", 2001.
 [2] 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.
 [3] Jari Korva, Johan Plomp, Peter Määttä, Meija Mesto "On-line Service adaptable for mobile and fixed terminal devices", 2000.

[4] Peter Herzum, Oliver Sims, "Business Component Factory", Wiley Computer Publishing, 2000.
 [5] Stephanidis, C. Concept of Unified User Interfaces, In Interfaces for All Concepts, Method and Tools, LAurence Erlbaum Associates, 2001.
 [6] 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.
 [7] "An Introduction to Web Clipping and PQAs", Eric Cook, <http://www.webreference.com/dev/webclip/index.html>
 [8] <http://coldjava.hypermart.net/servlets/wmltags.htm>
 [9] <http://coldjava.hypermart.net/servlets/escape.htm>



김 치 수

e-mail : cskim@kongju.ac.kr
 1984년 중앙대학교 컴퓨터공학과(학사)
 1986년 중앙대학교 컴퓨터공학과(석사)
 1990년 중앙대학교 컴퓨터공학과(박사)
 1990년~1992년 공주교육대학교 전임강사
 1992년~현재 공주대학교 컴퓨터공학부 교수

관심분야 : 소프트웨어 공학, 개발 방법론



김 영 태

e-mail : zerot@kongju.ac.kr
 2000년 공주대학교 전자계산학과(학사)
 2002년 공주대학교 전자계산학과 (이학석사)
 2002년~현재 공주대학교 컴퓨터공학과 박사과정

관심분야 : 객체지향시스템, 소프트웨어 개발방법론