

# 홈 네트워크 환경을 위한 적응형 사용자 인터페이스 설계 및 구현

정재환<sup>†</sup> · 장현수<sup>\*\*</sup> · 고광선<sup>\*\*\*</sup> · 김구수<sup>\*\*\*\*</sup> · 엄영익<sup>\*\*\*\*\*</sup>

## 요약

최근 홈 네트워크 환경에 PC, 노트북, PDA, 모바일 폰등과 같은 모바일 기기를 이용하여 다양한 기기를 제어하기 위한 연구와 개발이 활발해지고 있다. 모바일 기기들의 휴대성 및 실용성등의 장점을 이용하여 시간과 장소에 구애없이 가정 내의 네트워크로 연결된 디지털 기기들을 로컬 내지 원격에서 제어 및 관리가 가능하다. 그러나 이러한 모바일 기기들의 사용자 인터페이스는 서로 다른 언어를 사용하며, 기기별 인터페이스의 형태 및 사용법의 차이점으로 인한 어려움이 발생한다. 이러한 어려움을 해결하기 위해서는 크게 두 가지 고려사항이 존재한다. 첫째, 다양한 이기종 기기들을 제어하는데 필요한 사용자 인터페이스의 개발 및 유지보수의 어려움을 해결해야 한다. 두 번째, 사용자의 선호도 및 장치의 특징이 동적으로 반영된 사용자 인터페이스를 제공할 수 있어야 한다. 본 논문에서는 이러한 고려사항을 충족시키기 위하여 XML(eXtensible Markup Language) 기반의 UIML(User Interface Markup Language)과 사용자 및 장치에 대한 프로파일 정보를 이용하여 홈 네트워크 환경을 위한 적응형 사용자 인터페이스를 설계한 내용을 제시한다. 또한 TV, 전등, 프로젝터를 PC와 노트북, 모바일폰을 이용하여 제어할 수 있도록 구현한 내용을 보인다.

키워드 : 사용자 인터페이스, UIML, 프로파일, 트랜스코더, 매핑 규칙

## Design and Implementation of an Adaptive User Interface for Home Network Environments

Jae Hwan Jung<sup>†</sup> · Hyun Su Jang<sup>\*\*</sup> · Kwang Sun Ko<sup>\*\*\*</sup> · Gu Su Kim<sup>\*\*\*\*</sup> · Young Ik Eom<sup>\*\*\*\*\*</sup>

## ABSTRACT

With the growing proliferation of mobile computing devices, several industrial and academic research groups have a vigorous studying to remote control for various appliances with mobile devices such as Notebooks, PDAs, and Smartphones in home network environments. We can utilize the good points of mobile devices such as portability and usability so that we can remote control and manage the mobile devices connected on home networks anytime, anywhere. However, mobile devices use different languages. Therefore, they cause some problems because their interfaces and the methods of operation are different each other. To solve these problems, there are two consideration. First, we may be solved development of the user interface and difficulty of maintenance in order to control various heterogeneous devices. Second, we may provide the user interface which is dynamically adapting user's preferences and device characteristics. To satisfy these considerations, we describe the design of an adaptive user interface for home network environments using the UIML (User Interface Markup Language) based on XML (eXtensible Markup Language) and the profile information of the user and device. Therefore, we present several implementation examples that show how the framework can form the basis of prototypical applications.

Key Words : User Interface, UIML, Profile, Transcoder, Mapping Rule

## 1. 서론

IT 기술의 급속한 발달과 초고속망의 보급으로 홈 네트

워크 산업과 관련기기 시장에 대한 관심이 높아지고 있다. 홈 네트워크는 초고속 인프라를 기반으로 네트워크, 정보처리 등 다양한 IT 기술이 접목되어 있는 분야이다. 홈 네트워크 환경에서는 가정 내의 정보가전 기기가 네트워크로 연결되어 시간과 장소에 상관없이 다양한 서비스를 제공할 수 있다. 최근에는 홈 네트워크 환경에 상황인지 기술이 접목되어 사용자를 감지하고 서비스를 동적으로 제어 할 수 있

† 정 회 원 : (주) CSPI 시스템 엔지니어  
 \*\* 준 회 원 : 성균관대학교 대학원 전기전자및컴퓨터공학과 박사과정  
 \*\*\* 정 회 원 : 성균관대학교 대학원 이동통신공학과 연구교수  
 \*\*\*\* 정 회 원 : 동양대학교 정보통신공학부 교수  
 \*\*\*\*\* 종신회원 : 성균관대학교 정보통신공학부 교수  
 논문접수: 2007년 4월 11일, 심사완료 : 2007년 11월 16일

는 유비쿼터스 홈이라는 새로운 패러다임으로 변화해 가고 있다. 이더넷, 전화선, 무선, 전력선 등으로 네트워크화 되어 가정 내 디지털 기기들 간의 기능 및 데이터 공유, 원격 제어 등을 가능하게 한다.

홈 네트워크 환경의 발전과 더불어 모바일 기기에 대한 관심이 증가하고 있다. PDA, 모바일폰, 스마트폰 등 과거부터 사용되었던 기기들은 성능이 더욱 향상되고 있으며 UMPC, PMP 등의 새로운 기기가 등장하고 있다. 이러한 모바일 기기들은 휴대가 용이하며, 뛰어난 성능으로 다양한 분야에서 활용가치가 높다. 이러한 모바일 기기들의 여러 장점들을 다양한 IT 기술이 접목되어 있는 홈 네트워크 환경에 적용하면 여러 가지 장점을 얻을 수 있다. 첫째, 여러 종류의 모바일 기기를 이용하여 가전기기들의 원격 제어가 가능하다. 둘째, 공간적 제약 및 시간적 손실을 해결할 수 있다. 그러나 홈 네트워크 환경에 다양한 모바일 기기들의 특성을 접목하기 위해서는 몇 가지 고려해야 할 사항이 있다. 첫째, 여러 기기들의 이질적인 시스템 환경에서 사용 가능한 언어가 필요하다. 장치의 환경에 적합한 인터페이스를 개발하기 위해서는 여러 언어와 개발자를 필요로 하며 이는 시간적, 비용적 측면에 있어 많은 제약점을 유발한다. 이와 같은 제약사항을 해결하기 위해서 보편적인 언어를 이용하여 인터페이스를 개발하고 각 장치에 적합하도록 소스 코드를 변환하여 인터페이스를 생성하는 기법이 필요하다. 둘째, 이러한 언어를 누구나 쉽게 사용할 수 있어야 한다. 셋째, 사용자에게 최적화된 인터페이스를 제공하기 위해서 동적으로 사용자의 정보를 처리할 수 있는 방안이 필요하다.

최근 위와 같은 고려사항을 해결하기 위한 다양한 방법들이 대두되고 있다. IBM의 UIA(Universal Information Appliance) 프로젝트[1]는 정보 접근을 위한 사용자 인터페이스의 생성을 위해 MoDAL이라 불리는 XML기반의 언어를 사용한다. 그러나 UIA의 MoDAL 프로세서는 간단한 레이아웃과 텍스트 형식의 입력 값만이 처리 가능하다. 스탠포드 대학의 ICrafter[2]는 유비쿼터스 환경을 위한 서비스 프레임워크로써 사용자가 다양한 멀티모달을 이용할 수 있다. 이로써 단일모달을 사용하는 것에 비해 정확한 표현이 가능하여 보다 유연하게 장치와 상호작용이 가능하다. 그러나 ICrafter는 인터페이스의 생성방법에 있어 자동적인 방법보다 수동적인 방법에 더 초점을 두고 있다. UIA와 ICrafter는 사용자 인터페이스 생성을 위한 방안에만 주안점을 두고 있어, 사용자에게 개인화된 인터페이스를 제공하기 위한 사용자 및 장치의 정보 처리에 관한 사항은 다루지 않고 있다.

본 논문에서는 앞서 제시한 고려사항 및 위와 같은 문제점들을 해결하기 위해서 장치 독립적인 인터페이스의 생성에 목적이 있는 UIML에 특징에 사용자 및 장치의 특성 정보를 반영한 적응형 사용자 인터페이스를 전개하기 위한 방안을 제안한다. 적응형 사용자 인터페이스란 장치 독립적이며, 사용자의 선호도 정보와 인터페이스를 전개할 장치의 특성 정보에 따라 동적으로 변환되는 인터페이스로 정의된다.

본 논문에서 제안하는 방안은 UIML이라는 장치 독립적인 언어를 사용하여 플랫폼에 독립적이다. 또한 XML언어를 기반으로 하여 프로그래머뿐만 아니라, 일반 사용자도 쉽게 인터페이스를 개발할 수 있다. 새로운 DTD나 XML 스키마를 정의하여 다양한 방면에서 응용이 가능하며, 간단한 구조로 구성되어 있어 쉽게 확장이 가능하다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 인터페이스 개발을 위한 XML 기반의 언어들에 대해 기술하고, 제 3 장에서는 본 논문에서 제안하는 적응형 사용자 인터페이스의 생성을 위한 시스템 구조 및 프로파일 구조에 대해 설명한다. 제 4 장에서는 프로파일과 UIML 인터페이스 파일의 접목을 위한 매핑규칙에 대해 설명하고, 제 5 장에서는 적응형 사용자 인터페이스를 가상의 홈 네트워크 환경에서 구현한 내용을 보여준다. 마지막으로 제 6 장에서 결론을 맺는다.

## 2. 인터페이스 개발을 위한 XML기반의 언어

본 장에서는 플랫폼과 프로그래밍 언어에 독립적이고, 간단한 구조로 쉽게 사용가능한 XML 기반 인터페이스 개발 언어에 대해 소개한다.

### 2.1 UIML(User Interface Markup Language)

UIML은 여러 디지털 기기들의 이기종성으로 인해 발생하는 다양한 문제를 해결하기 위해 등장한 언어이다. UIML은 하나의 범용적인 언어를 통해 여러 기기들의 사용자 인터페이스를 구축하여 개발기간의 단축, 원활한 유지보수를 목적으로 한다[3][4]. 다음은 UIML의 메타-인터페이스 모델을 구성하는 요소의 특징이다.

- 인터페이스: 인터페이스의 실제 구조와 내용을 기술하는 부분.
- 프리젠테이션: 사용자 인터페이스를 변환할 언어의 태그를 기술하는 부분.
- 로직: 인터페이스를 구현하는 애플리케이션의 논리 구조를 기술하는 부분.

UIML 문서의 기본 구조는 XML의 기본 문법을 따르며 루트 엘리먼트로 <uiml> 태그를 사용한다. 그리고 자식 엘리먼트로 헤드, 인터페이스, 템플릿, 피어를 포함한다. 헤드 엘리먼트는 UIML문서에 관련된 메타 데이터를 제공한다. 인터페이스 엘리먼트는 사용자 인터페이스를 구성하는 구조, 콘텐츠, 스타일, 행위 엘리먼트로 세분화된다. 구조 엘리먼트는 여러 플랫폼을 위한 인터페이스 파트들과 구성요소를 기술한다. 콘텐츠 엘리먼트는 사용자에게 전달할 워드, 이미지, 사운드 등의 정보를 가지고 있다. 스타일 엘리먼트는 인터페이스와 관련된 다양한 속성 값들을 기술한다. 행위 엘리먼트는 인터페이스 이벤트들과 실행할 동작들이 무엇인지를 기술한다. 템플릿 엘리먼트는 UIML 코드의 부분들을 재사

용을 가능하게 해 준다. 피어 엘리먼트는 UIML의 각 속성과 이벤트 이름을 UI 톨킷과 내부 로직에 매핑시키는 역할을 한다.

### 2.2 XUL(XML-based User Interface Language)

XUL은 모질라 웹 브라우저를 보다 빠르고, 쉽게 개발하기 위해 설계된 XML기반의 언어이다[5]. XUL은 HTML과 다른 XML을 포함할 수 있으며, 사용자 인터페이스의 모양을 변경하기 위해 스타일 시트를 사용한다. 사용자 인터페이스는 컨텐츠, 스킨, 로컬의 세부분으로 구성되어 있다. 컨텐츠는 사용자 인터페이스 요소들과 스크립트를 포함하며, 스킨은 스타일 시트와 이미지들을 포함하고 있다. 스타일 시트는 CSS를 이용하며, 윈도우의 모양을 보다 세밀하게 기술하기 위해 사용된다. XUL은 자바 스크립트를 이용하여 HTML에서 사용하던 방법과 유사하게 이벤트를 처리하기 위해 이벤트 핸들러를 사용한다.

### 3. 적응형 사용자 인터페이스를 위한 시스템 구조

본 장에서는 장치 독립적인 인터페이스를 생성하고, 동적으로 사용자의 관심 정보를 반영할 수 있는 시스템 구조와 이를 구성하는 컴포넌트에 대해 기술한다.

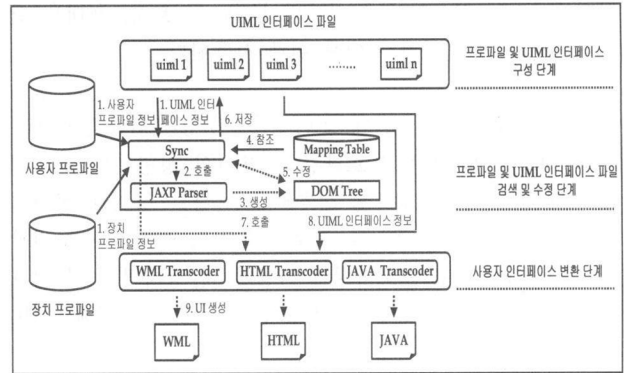
본 논문에서 제안하는 시스템의 기능 및 인터페이스의 목적은 다음과 같다.

- 기본적인 인터페이스의 용도에 부합된 기능만을 제공한다.
  - 기존의 인터페이스가 가지고 있는 불필요한 기능을 제거하여 복잡한 인터페이스에서 발생하는 불편함을 해소한다.
- 여러 장치에서도 공통적으로 사용가능한 인터페이스를 제공한다.
  - 공통의 인터페이스를 제공하여 각 장치별 제어를 위한 인터페이스에서 발생하는 혼돈성을 해결한다.
- 사용자의 선호도에 따른 인터페이스의 변경 기능을 제공한다.
  - 시스템이 정한 범위 내에서 인터페이스 변경 기능을 두어 사용자에게 개인화된 인터페이스를 사용할 수 있도록 한다.

#### 3.1 적응형 사용자 인터페이스 생성을 위한 시스템 구조

본 장에서는 본 논문에서 제안하는 적응형 사용자 인터페이스 생성을 위한 시스템 구조와 컴포넌트의 특징, 그리고 각 단계별 발생하는 절차에 대해 설명한다.

(그림 1)에서 제시한 시스템은 크게 3단계로 구성되며, 각 단계별 절차 및 컴포넌트의 특징, 동작과정은 다음과 같다. 먼저 프로파일 및 UIML 인터페이스 구성 단계는 사용자 및 장치 프로파일 구조에 적합하도록 정보를 입력하고, UIML



(그림 1) 시스템 구조

표준문서에 맞도록 사용자 인터페이스를 구성하는 단계이다. 사용자 및 장치 프로파일의 생성을 위해서는 본 논문에서 제시하고 있는 프로파일 구조에 맞도록 입력을 해야 한다. 프로파일의 구조 및 특징에 관한 내용은 본 논문의 3.2절에서 상세히 다룬다. UIML의 경우 UIML에서 제공하고 있는 XML 스키마, DTD의 수정을 통해 사용자 및 그룹의 특성과 목적에 따라 다양한 방향으로 확장이 가능하다 [6][7][8]. 본 논문에서는 텔레비전, 전등, 프로젝터를 제어하기 위한 UIML 인터페이스를 구성하였으며, 5장에서 인터페이스를 구현한 내용을 다룬다. 본 논문에서는 다양한 환경에서 플랫폼 독립적이며 동적인 사용자 인터페이스를 생성하기 위한 방안에 초점을 두고 있기 때문에, 프로파일 및 UIML 구조에 관한 내용은 제공하고 있지만, 작성을 위한 도구는 다루지 않는다.

두 번째로 프로파일 및 UIML 인터페이스 파일 검색 및 수정 단계는 프로파일에서 장치의 특징 및 사용자의 관심정보를 추출하여 작성된 UIML 인터페이스 파일에 반영하기 위한 단계이다. 프로파일과 UIML 인터페이스 파일의 병합은 미리 정해진 매핑 규칙에 따라 진행되며, DOM객체를 구현한 Sync와 JAXP 파서를 이용하여 프로파일의 정보를 검색하여 작성된 UIML 인터페이스 파일에 반영한다. 매핑 규칙에 관한 내용은 본 논문의 4장에서 상세히 다룬다. Sync는 XML로 작성된 프로파일 및 UIML 인터페이스 파일을 읽어드리고, 해석을 위해 XML 파서인 JAXP 파서를 호출한다. JAXP 파서는 태그명, 속성명, 속성값 및 엘리먼트 내용의 분리등 문서를 해석 한 후 DOM 트리를 생성한다.

Sync는 각 문서의 정보 매핑을 위해 매핑 테이블을 참조하여 생성된 DOM 트리의 내용을 검색 후, 프로파일의 정보를 UIML 인터페이스 파일에 반영한다. 정보의 매핑이 완료 되면 Sync는 사용자 인터페이스 변환을 위해 트랜스코더를 호출한다.

마지막으로 트랜스코더를 이용한 사용자 인터페이스 변환 단계에서는 개인화된 인터페이스를 트랜스코더를 이용하여 사용자가 요구한 장치에 사용하는 코드로 변환하는 단계이다. Sync로부터 사용자 인터페이스 변환을 위한 요청이 발생할 경우 트랜스코더는 UIML로 작성된 인터페이스 파일의 내용을 분석하여, 목적에 맞는 언어로 변환한다. 변환된

인터페이스는 사용자에게 전달되어, PC, 노트북, 휴대용 단말기 등에서 가전기기 제어를 위해 사용된다.

3.1.1 프로파일 및 UIML 인터페이스 구성을 위한 컴포넌트

- **사용자 및 장치 프로파일:** 사용자의 성명, 주민등록번호 등의 신상정보 및 선호도와 관련된 정보와 장치의 모델, 제조사, 해상도 정보 등의 장치 특성정보가 저장된 파일이다. 프로파일은 사용자 및 장치의 특성 정보가 반영된 인터페이스의 생성과 확장성을 위해 XML 기반으로 구성하였다.
- **UIML 인터페이스 파일:** UIML 작성 도구나 기타 편집기를 이용하여 UIML 문법에 맞도록 작성된 소스 코드이다. 사용자 인터페이스는 기본적으로 작성된 UIML 인터페이스에 사용자 및 장치 프로파일의 정보를 반영하여 개인화된 인터페이스를 제공한다.

3.1.2 프로파일 및 UIML 인터페이스 파일 검색 및 수정을 위한 컴포넌트

- **Sync:** 사용자 및 장치 프로파일의 정보를 UIML 인터페이스에 반영하기 위해 DOM 객체를 구현한 프로세스이다. UIML은 장치 독립적인 인터페이스의 생성에 목적이 있기 때문에 사용자 및 장치의 정보를 반영하기 위해서는 별도의 방안을 필요로 한다. 본 논문에서는 Sync라는 프로세스를 구현하여, 이러한 문제점을 해결하였다.

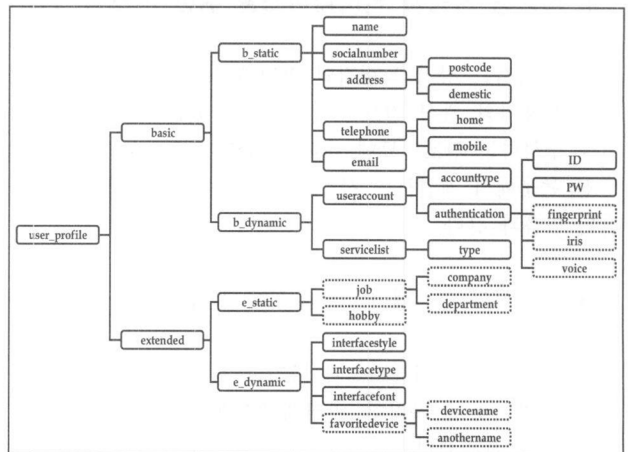
3.1.3 사용자 인터페이스 변환을 위한 컴포넌트

- **WML 트랜스코더:** 전화기와 같은 휴대용 단말기에서 인터페이스를 전개하기 위해 UIML로 작성된 사용자 인터페이스를 WML로 변환하여 주는 모듈이다. Sync에 의해 WML 트랜스코더 호출 명령을 통해 UIML 소스 코드를 분석하고 UIML의 엘리먼트를 WML 엘리먼트로 변환하여 WML 소스 코드를 생성한다.
- **HTML 트랜스코더:** 웹브라우저에 인터페이스를 전개하기 위해 UIML로 작성된 사용자 인터페이스를 HTML로 변환하여 주는 모듈이다. Sync에 의해 HTML 트랜스코더 호출 명령을 통해 UIML 소스 코드를 분석하고 UIML의 엘리먼트를 HTML 엘리먼트로 변환하여 WML 소스 코드를 생성한다.
- **JAVA 트랜스코더:** 자바 플랫폼에서 인터페이스를 전개하기 위해 UIML로 작성된 사용자 인터페이스를 JAVA로 변환하여 주는 모듈이다. Sync에 의해 JAVA 트랜스코더 호출 명령을 통해 UIML 소스 코드를 분석하고 UIML의 엘리먼트를 JAVA 클래스로 변환하여 WML 소스 코드를 생성한다.

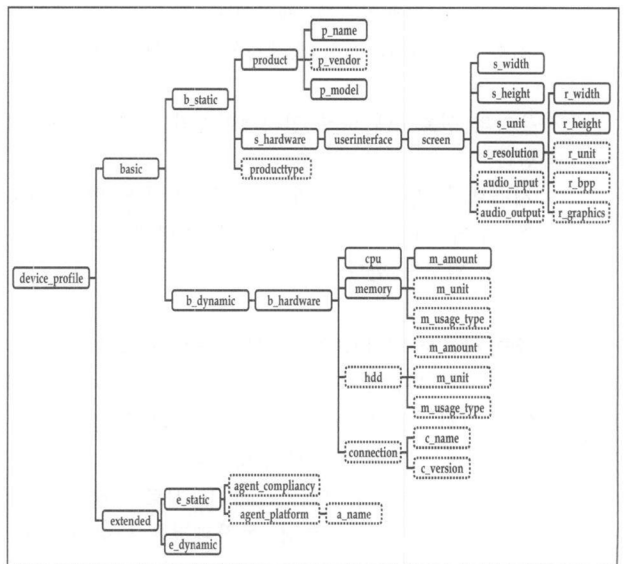
3.2 사용자 및 장치 프로파일 구조

본 장에서는 개인화된 인터페이스를 생성하기 위해 필요

한 사용자 및 장치의 정보관리를 위한 프로파일의 구조 및 특성에 대해 설명한다. UIML은 하나의 범용적인 언어를 통해 여러 기기들의 사용자 인터페이스를 구축하여 개발기간의 단축, 원활한 유지보수를 목적으로 한다. 이러한 특성으로 인해 장치 독립적인 인터페이스의 생성은 가능하나, 사용자의 개인화 정보와 장치의 특성과 관련된 부분은 다루지 않고 있다. 본 논문에서는 XML 기반의 사용자 및 장치 프로파일 구조를 생성하여 이러한 제약사항을 해결한다. (그림 2)와 (그림 3)은 사용자 및 장치 프로파일을 트리구조로 나타낸 것이다. 프로파일은 플랫폼과 프로그래밍 언어에 독립적이고, 다양한 환경에서의 확장성을 고려하여 XML 기반으로 구성하였다. 사용자 및 장치 프로파일의 정보들은 장치의 특징과 사용자의 관심정보를 UIML 인터페이스에 반영하기 위한 것으로 Sync에 의해 참조된다. 사용자 프로파일과 장치 프로파일은 기본정보(basic), 확장정보(extended), 정적정보(static), 동적정보(dynamic)의 4가지 정보로 구성되며, 동일한 구조를 가진다.



(그림 2) 사용자 프로파일 구조



(그림 3) 장치 프로파일 구조



(그림 2)에서 제시한 사용자 프로파일은 다양한 환경을 지원하기 위해 포탈, 은행, 쇼핑몰 등의 웹 사이트에서 회원 가입 시 요구하는 정보와 사용자 선호도 정보를 접목시켜 구성하였다. (그림 3)에서 제시한 장치 프로파일은 CC/PP (Composite Capability/Preference Profiles)[9]에서 제시되어 있는 항목중에서 옵션과 같은 불필요한 부분을 제외하여 본 시스템에 적합하도록 재구성하였다. (그림 2)와 (그림 3)의 구조에서 실선으로 표시된 부분은 반드시 입력되어야 하는 값을 의미하며, 점선으로 표시된 부분은 선택을 의미한다. 기본정보는 직관적으로 표현 가능한 정보(예: 사용자 이름, 나이, 장치 이름, 제조사)를 분류하기 위해 사용된다. 확장정보는 직관적으로 사용자나 장치를 나타내기 어려운 정보들(예: 사용자 취미, 직업, 선호도, 장치의 에이전트 지원여부)을 분류하기 위해 사용된다. 정적정보는 사용자의 이름, 주민등록번호, 장치의 이름, 제조사, 모델명, 화면크기, 해상도 등 고정적인 정보에 해당하는 사항들로 구성된다. 동적정보는 사용자의 취미, 선호도(인터페이스 스타일, 타입), 계정정보(ID/Password)등과 같이 정보의 변경이 주기적 혹은 정기적으로 발생 가능한 사항들로 구성된다. 이와 같이 각 정보들을 특성에 맞도록 분류함으로써 정보의 검색 및 수정이 용이하며, 정보의 추가 시에도 효율적이다. 본 논문에서는 동적인 사용자 인터페이스 생성에 관해 주안점을 두었기 때문에 프로파일의 추가 및 삭제와 관련된 사항에 대해서는 사용자의 수동적인 입력 및 삭제를 필요로 한다.

#### 4. 프로파일과 UIML 인터페이스 파일 간 병합을 위한 매핑 규칙

본 논문에서 제시한 시스템 구조의 프로파일 및 UIML 인터페이스 파일 검색 및 수정 단계에서 Sync와 JAXP 파서가 사용자 및 장치 프로파일의 내용과 UIML 인터페이스 파일의 내용을 병합하기 위해서는 정형화된 규칙이 필요하다. 정형화된 규칙을 생성하기 위해서는 프로파일과 UIML 인터페이스를 구성하는 엘리먼트들 중에서 사용자 선호도 및 장치의 해상도 정보와 같이 동적인 특징과 관련된 정보를 추출해야 한다. 그리고 추출된 엘리먼트들에 대해 지정된 값을 부여하는 것이 필요하다. 본 논문에서는 추출된 엘리먼트에 정형화된 값을 지정할 매핑 테이블을 생성하여 Sync가 프로파일의 내용을 UIML 인터페이스 파일에 저장 시 참조하도록 하였다. 다음 표 1은 사용자 및 장치 프로파일에서 추출된 동적인 인터페이스와 관련된 엘리먼트에 대한 내용이다.

<표 1>에서 개인화와 관련된 사용자 프로파일의 엘리먼트는 인터페이스의 종류, 스타일, 폰트, 타입의 사용자 선호도와 관련된 엘리먼트이다. 장치 프로파일의 엘리먼트는 장치의 화면크기, 해상도 등 장치의 특성과 관련된 엘리먼트이다. 최종적으로 사용자에게 전달되는 인터페이스는 <표 1>의 프로파일의 엘리먼트에 저장되어 있는 값에 따라 인터페이스의 형태(크기, 스타일, 폰트)가 달라진다. 모든 사용자

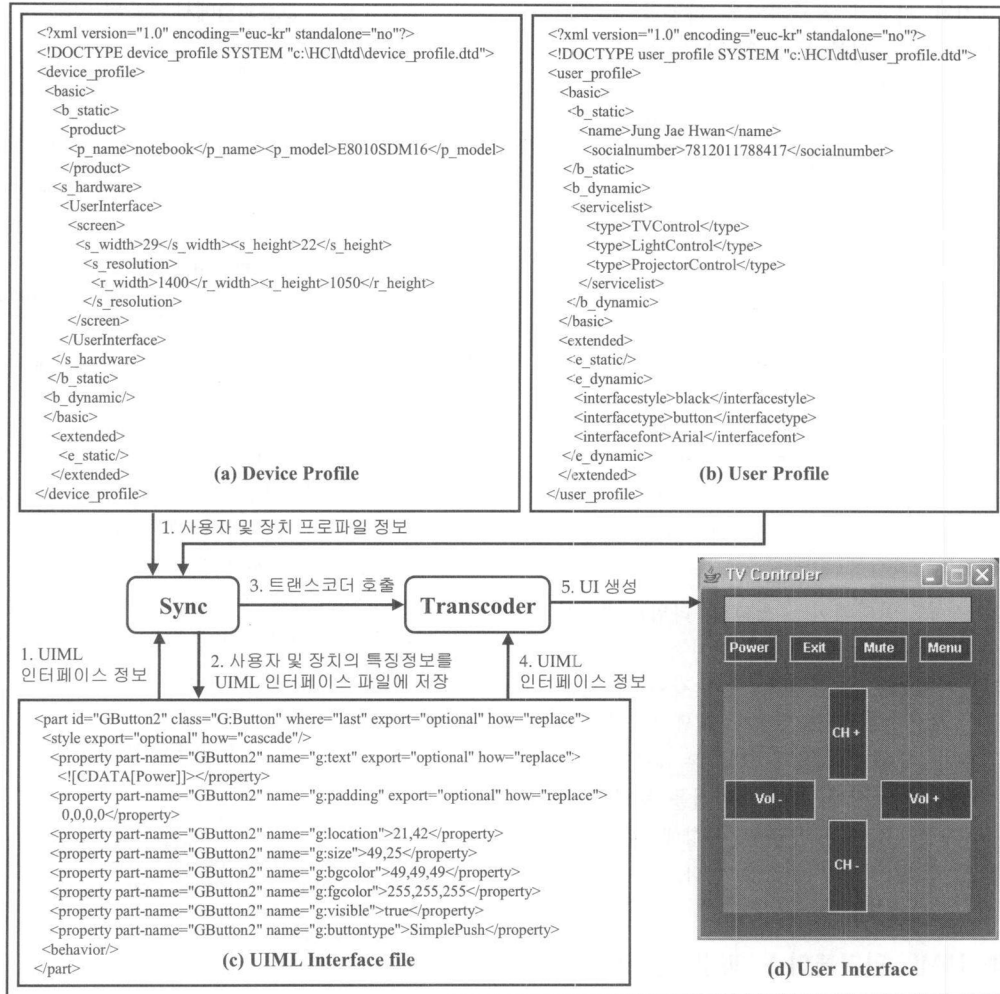
<표 1> 프로파일에서 추출된 엘리먼트

순번	사용자 프로파일의 엘리먼트	장치 프로파일의 엘리먼트
1	basic.b_dynamic_servicelist_type	basic.b_static.s_hardware.screen.s_height
2	extended.e_dynamic.interfacestyle	basic.b_static.s_hardware.screen.s_width
3	extended.e_dynamic.interfacefont	basic.b_static.s_hardware.screen.s_resolution.r_width
4	extended.e_dynamic.interfacefonttype	basic.b_static.s_hardware.screen.s_resolution.r_height

의 선호도에 따른 다양한 인터페이스의 형태(스타일, 색깔, 폰트)를 지원하기란 사실상 매우 어렵다. 그래서 본 시스템에서는 엘리먼트의 값들에 대해 한정을 지었다. 즉, 시스템이 정한 범위 내에서 사용자는 자신이 원하는 형태의 다양한 인터페이스를 가질 수 있다. 다음 <표 2>는 사용자 및 장치 프로파일과 UIML 인터페이스 파일의 엘리먼트 및 값에 대한 매핑 테이블의 예이다.

<표 2> 매핑 테이블의 예

user profile		value			
		user 1	user 2	user 3	
b_dynamic_servicelist_type		television	projector	light	
extended.e_dynamic_interfacestyle		black	blue	green	
extended.e_dynamic_interfacefont		Arial	Comic Sans	Century Gothic	
extended.e_dynamic_interfacefonttype		button	button	button	
device profile		value			
b_static.s_hardware.screen.s_height		25	28	16.3	
b_static.s_hardware.screen.s_width		38	34	21.8	
b_static.s_hardware.screen.s_resolution.r_width		1450	1280	800	
b_static.s_hardware.screen.s_resolution.r_height		1050	1024	480	
partid	style		value		
	part-name	name			
GButton2	GButton2	g:text	Power	Power	Power
		g:location	21,42	21,42	21,42
		g:size	49,25	46,22	44,20
		g:bgcolor	49,49,49	255,255,255	0,128,0
		g:fgcolor	255,255,255	0,0,0	0,0,0
		g:font	Arial	Comic Sans	Century Gothic



(그림 4) 사용자 인터페이스 생성 과정

UIML 언어가 가지고 있는 목적 언어로의 변환에 대한 어휘적 제약사항으로 인해 인터페이스에 사용자 선호도와 장치의 특성을 반영할 수 있는 부분은 한정되어 있다. <표 2>는 텔레비전 제어를 위한 인터페이스로 인터페이스의 구성 요소인 전원버튼의 속성(크기, 위치, 폰트등)이 사용자 및 장치 프로파일의 내용에 따라 가지는 값을 나타낸 것이다. GButton2의 속성 중 g:bgcolor, g:fgcolor의 값은 사용자 프로파일의 interfacestyle 엘리먼트에 입력된 값에 의해 결정되며, g:font는 interfacefont 엘리먼트에 입력된 값에 의해 결정된다. 시스템에서는 사용자가 interfacestyle, interfacefont 값을 지정하지 않을 경우 기본적으로 black, Arial 폰트로 인터페이스를 제공한다. 장치의 s\_height, s\_width, r\_width, r\_height 엘리먼트에 입력되어 있는 값에 의해 인터페이스의 크기와 위치의 속성값이 결정된다. 즉, 사용자 및 장치 프로파일의 엘리먼트에 입력된 값에 따라 인터페이스의 속성값이 달라진다. 이로써 사용자는 시스템이 정한 범위 내에서 프로파일의 수정을 통해 다양한 인터페이스를 자신의 선호도와 장치에 맞게 변경된 인터페이스를 사용할 수 있다.

아래 (그림 4)는 사용자 인터페이스 생성 과정을 나타낸 것으로 (그림 4) (a)는 제품명, 모델명, 화면의 크기, 해상도

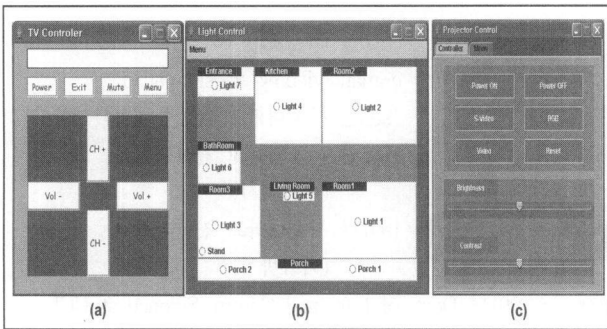
에 관한 정보가 입력되어 있는 장치 프로파일을 나타낸 것이다. (그림 4) (b)는 사용자 이름, 주민번호, 서비스목록, 인터페이스 스타일, 폰트등에 관한 정보가 입력되어 있는 사용자 프로파일을 나타낸 것이다. (그림 4) (c)는 Sync에 의해 사용자 및 장치 프로파일에서 추출된 엘리먼트의 값이 반영된 UIML 인터페이스 파일의 내용을 나타낸 것이다. XML기반의 사용자 및 장치 프로파일과 UIML 인터페이스 파일은 Sync에 의해 호출되는 JAXP 파서에 의해 해석된다. Sync는 JAXP 파서에 의해 해석된 내용을 바탕으로 (그림 1)에서 나타난 엘리먼트의 내용을 추출하여, UIML 인터페이스 파일의 각 파트가 가지는 속성을 변경한다. 사용자 및 장치 프로파일의 특징을 UIML 인터페이스 파일에 적용하는 과정이 끝나면, UIML 인터페이스 파일은 각 목적에 맞는 트랜스코더에 의해 해석되어 최종 사용자에게 전달될 인터페이스가 생성이 된다. (그림 4) (d)는 위에서 설명한 모든 과정을 거친 JAVA 기반의 사용자 인터페이스를 나타낸 것이다.

### 5. 구현 및 평가

본 장에서는 3장과 4장에서 제시한 적응형 사용자 인터페

<표 3> 구현 환경

환경	내용
운영체제	WindowsXP Home Edition
프로세서	Intel Pentium Mobile processor 1.60GHz
인터페이스 설계도구	Harmonia LiquidUI Version 4.0 beta 10
	Altova XML Spy 2007 Enterprise Edition Trial
	Borland JBuilder 2005 Enterprise Trial EditPlus2 Trial
XML 파서	Sun microsystems JAXP 1.2.4 package
시뮬레이터	Openwave Phone Simulator Version 7.0
Java SDK	J2SE 1.4.2_12



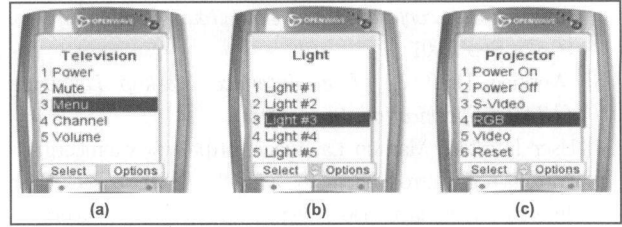
(그림 5) JAVA 기반의 적응형 사용자 인터페이스

이스의 생성을 위한 시스템 구조와 매핑 규칙을 기반으로 홈 네트워크 환경에서 가전기기를 제어하는 인터페이스를 구현한 내용을 설명한다. 구현 환경은 <표 3>과 같다.

홈 네트워크 환경의 모든 가전기기를 제어하는 인터페이스 개발은 많은 어려움이 있기 때문에 제어 가능한 가전기기를 텔레비전, 전등, 프로젝터로 한정하였다. 인터페이스의 설계는 자주 사용되는 기능만을 나타낸 간단한 구성에 중점을 두었다. 이로써 복잡한 인터페이스에서 발생하는 어려움을 해소하며, 효율적으로 동적인 인터페이스를 제공할 수 있다. 한 장치에서 다른 인터페이스로 전환할 수 있는 기능을 추가하여, 별도의 과정 없이 다양한 가전기기를 제어할 수 있도록 하였다. 사용자 인터페이스는 기본적으로 black 스타일, Arial 폰트로 제공이 되며 blue 및 green 스타일, Comic Sans과 Century Gothic 폰트는 기본적으로 제공되는 인터페이스의 각 엘리먼트를 변환하여 제공된다.

(그림 5)는 JAVA 기반의 사용자 인터페이스로 사용자 및 장치 프로파일의 엘리먼트의 값에 따라 다양하게 변화된 것을 나타낸 것이다. (그림 5) (a)는 버튼 타입의 블루스타일, Comic Sans MS-Plain체로 변환된 텔레비전 제어를 위한 인터페이스이다. (그림 5) (b)는 버튼타입, 블랙스타일, Arial-Bold체로 변환된 전등제어를 위한 인터페이스이다. 사용하기 쉽도록 집안의 구조와 유사한 형태를 가진다. (그림 5) (c)는 버튼타입, 그린스타일, Arial-Bold체로 변환된 프로젝터 제어를 위한 인터페이스이다.

(그림 6)는 WML기반의 사용자 인터페이스를 나타낸 것이다. (그림 6) (a)는 텔레비전 제어를 위한 인터페이스이고, (그림 6) (b)는 전등 제어를 위한 인터페이스, (그림 6) (c)



(그림 6) WML 기반의 적응형 사용자 인터페이스

는 프로젝터 제어를 위한 인터페이스이다. WML기반의 경우 휴대폰과 같은 단말기의 성능상의 제약성으로 인해 GUI 표현이 제한되어 있다. 각각의 기능은 단축버튼 또는 위치 이동 버튼을 이용하여 선택이 가능하다.

이렇게 생성된 인터페이스는 장치 독립적으로 사용이 가능하며, 사용자 측면에서는 자신의 선호도가 반영된 개인화된 인터페이스를 사용할 수 있으며, 개발자적 측면에서는 하나의 보편적인 언어를 통해 인터페이스를 개발함으로써 시간적, 비용적인 면에서 많은 장점이 있다.

## 6. 결 론

홈 네트워크 환경과 같은 다양한 컴퓨팅 장치들이 상호 연결되어 있는 환경에서는 다양한 문제점이 발생한다. 이질적 시스템 환경에서 발생하는 장치 간 상호호환성문제, 장치에서 사용되는 인터페이스 개발의 어려움, 사용자의 관심 정보 등이 대표적이다. 본 논문에서는 이러한 문제점들을 해결하기 위해 장치 독립적인 언어인 UIML과 프로파일 기반의 사용자 및 장치 정보를 혼합하여 사용자 인터페이스를 생성하는 방안에 대해 제시하였다.

또한 제시한 기법을 이용하여 홈 네트워크 환경에서 다양한 컴퓨팅 장치로 가전기기를 제어하는 인터페이스를 구현하였다. 본 논문에서 제시한 시스템 구조는 XML 기반으로 구성되어 다양한 환경에 유연하게 대응할 수 있는 확장성을 제공한다. 또한 이기중 플랫폼에서 보편적으로 사용 가능한 언어를 사용하여, 장치 독립적인 인터페이스의 생성 및 제공이 가능하다. 사용자 인터페이스는 사용자의 관심 정보, 장치의 특성이 반영되어 사용자에게 개인화된 인터페이스를 제공하도록 하였다. 또한 간단한 사용자 인터페이스의 구성을 통해 복잡하고 서로 상이한 인터페이스에서 발생하는 사용상의 어려움과 생소함을 해결하였다.

## 참 고 문 헌

- [1] K. F. Eustice, T. J. Lehman, A. Morales, M. C. Munson, S. Edlund and M. Guillen, "A Universal Information Appliance," *IBM Systems Journal*, Vol. 38, pp. 575-601, 1999.
- [2] S. R. Ponnekanti, B. Lee, A. Fox, P. Hanrahan and T. Winograd, "ICrafterICrafter: A Service Framework for Ubiquitous Computing Environments", *Proc. of the 3rd*

*international conference on Ubiquitous Computing*, pp. 56-75, Sep. 2001

[3] Working Draft 3.1, *User Interface Markup Language (UIML) Specification*, Mar. 2004.

[4] User Interface Markup Language, <http://www.uiml.org>.

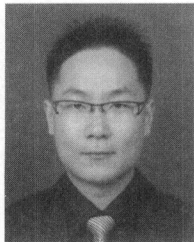
[5] Mozilla.org, Introduction to a XUL (XML-based User Interface Language) Document, <http://www.mozilla.org>.

[6] M. Abrams, C. Phanouriou, A. L. Batongbacal, S. M. Williams and J. E. Shuster, "UIML: An Appliance-Independent XML User Interface Language", *Proc. of the World Wide Web Conference*, Toronto, May. 1999.

[7] M. Abrams and C. Phanouriou, "UIML: An XML Language for Building Device Independent User Interfaces", *Proc. of XML'99*, Philadelphia, Dec. 1999.

[8] M. F. Ali and M. Abrams, "Simplifying Construction of Multi-Platform User Interfaces Using UIML", *Proc. of UIML 2001*, France, Mar. 2001.

[9] *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*, W3C Recommendation, Jan. 2004.



**정재환**

e-mail : kinocjh@ece.skku.ac.kr  
 2004년 대구가톨릭대학교 컴퓨터공학과 (학사)  
 2007년 성균관대학교 대학원 컴퓨터공학과 (석사)  
 2003년~현재 (주) CSPI 시스템 엔지니어

관심분야: 유비쿼터스 컴퓨팅, HCI, 미들웨어 등

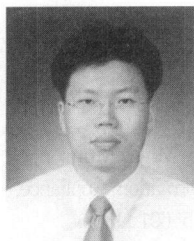


**장현수**

e-mail : jhs4071@ece.skku.ac.kr  
 2002년 성균관대학교 전기전자및컴퓨터 공학과(학사)  
 2005년 성균관대학교 대학원 전기전자및 컴퓨터공학과(공학석사)  
 2007년~현재 성균관대학교 대학원

전기전자및컴퓨터공학과 박사과정

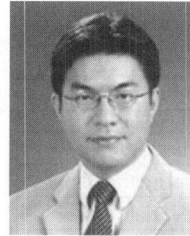
관심분야: 유비쿼터스 컴퓨팅, 이동 에이전트, HCI, 미들웨어 등



**고광선**

e-mail : rillar91@ece.skku.ac.kr  
 1998년 성균관대학교 정보공학과 (학사)  
 2004년 성균관대학교 대학원 전기전자 및 컴퓨터공학부(공학석사)  
 2007년 성균관대학교 대학원 전자전기 컴퓨터공학과(공학박사)

2007년~현재 성균관대학교 대학원 이동통신공학과 연구교수  
 관심분야: 정보보호, 리눅스, 네트워크 등



**김구수**

e-mail : gusukim@dyu.ac.kr  
 1994년 성균관대학교 정보공학과 (학사)  
 1996년 성균관대학교 정보공학과 (공학석사)  
 2006년 성균관대학교 정보통신공학과 (공학박사)

2007년~현재 동양대학교 정보통신공학부 교수  
 관심분야: 분산 컴퓨팅, 이동 에이전트 등



**엄영익**

e-mail : yieom@ece.skku.ac.kr  
 1983년 서울대학교 계산통계학과 (학사)  
 1985년 서울대학교 전산과학과 (이학석사)  
 1991년 서울대학교 전산과학과 (이학박사)

2000년~2001년 Dept. of Info. and Comm. Science at UCI  
 보호 방문교수

2004년~2005년 한국정보처리학회 논문지 편집이사  
 2005년~2006년 한국정보처리학회 학회지 편집위원장(상임이사)  
 1993년~현재 성균관대학교 정보통신공학부 교수  
 2007년~현재 성균관대학교 정보통신처 처장  
 관심분야: 분산 컴퓨팅, 이동 컴퓨팅, 이동 에이전트, 시스템 보안, 운영체제, 내장형 시스템 등