

통합리모컨의 태스크기반 사용자 인터페이스 생성을 위한 가전기기 마크업 언어

(Home Appliance Markup Language for Generating Task-Based User Interfaces of Universal Remote Controls)

임 성 수 [†] 조 성 배 ^{**}
(Sungsoo Lim) (Sung-Bae Cho)

요약 최근 각종 가전기기들이 유/무선을 통해 연결되어 하나의 서비스를 제공하고 있지만, 기기제어를 위한 인터페이스는 통합되지 않고 각각 존재한다. 따라서 사용자는 각 기기를 어떻게 제어해야 하는지를 숙지해야 하고, 기기의 기능들이 서로 어떻게 결합되어 서비스를 제공하는지 알아야 한다. 하지만 기기 제어를 위한 리모컨에는 기기의 복잡한 기능 제어를 위해 사용성이 높지 않은 다수의 버튼이 존재하며 각 기기별로 비슷한 모양의 리모컨들이 존재하므로, 하나의 서비스를 제공받기 위해 다수의 리모컨을 사용해야 하는 사용자에게 많은 어려움이 있다. 본 논문에서는 다양한 기기들이 서로 연결되어 제공되는 서비스 작업단위를 태스크로 정의하고, 사용자가 원하는 서비스를 손쉽게 제공받을 수 있도록 태스크 단위의 사용자 인터페이스 생성을 위한 가전기기 마크업 언어(Home Appliance Markup Language; HAML)를 제안한다. HAML은 태스크 제어에 꼭 필요한 버튼들만을 모아 하나의 인터페이스를 구성하고, 태스크 수행을 위한 기기 설정의 자동화 기능을 제공한다. 시스템 동작 시나리오와 사용성 평가를 통해 제안하는 방법의 유용성을 검증한다.

키워드 : 통합리모컨, 가전기기 마크업 언어, 태스크기반 인터페이스

Abstract Recently, various home appliances are inter connected in wired/wireless network to provide a single service, but their interfaces are not uniformly integrated. In order to get a service in such an environment, users have to figure out how to control each appliance and the way of combining its functions with others. Even worse to the users, many buttons are barely used to control the complicated functions of the appliances on the remote controllers, and controllers have very similar shapes in each other. In this paper, based on the definitions of a task as a unit of services provided in the environment of multi-connected appliances, we propose a markup language for home appliances, called HAML (Home Appliance Markup Language), for generating task-based user interfaces to help controlling multi-connected appliances as one pleases. The proposed method generates interfaces by gathering the buttons frequently used and necessary for tasks, and provides automatic settings of multi-connected appliances. The proposed method is verified with an analysis of scenarios and usability tests.

Key words : Universal remote controller, home appliance markup language, task-based interface

· 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업입금(No. 2009-0083838)

[†] 학생회원 : 연세대학교 컴퓨터과학과
lss@scslab.yonsei.ac.kr
^{**} 종신회원 : 연세대학교 컴퓨터과학과 교수
sbcho@yonsei.ac.kr
논문접수 : 2008년 12월 4일
실사완료 : 2009년 10월 21일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제37권 제1호(2010.1)

1. 서론

임베디드 시스템이 들어간 홈/정보 가전기기가 등장하면서 기존의 가전기기에는 없었던 다양한 기능들이 포함되고 있고[1], 다수의 기기가 서로 연결되어 서비스를 제공한다. 이에 따라 가전기기 제어를 위한 인터페이스는 더욱 사용하기 어렵게 되었다[2]. 대부분의 가전기기들은 복잡하고 유용한 기능들을 가지고 있지만, 리모컨 인터페이스의 공간적 제약을 극복하기 위한 효율적인 버튼 배치와 복잡한 기능을 수행하기 위한 여러 버튼의 사용 및 이해하기 힘든 버튼의 레이블 등으로 많

은 기능들이 사용되지 않는다[3].

태블릿 PC, 스마트 폰 등의 휴대 기기들이 등장하면서 이를 이용한 다중 기기 환경에서 사용자 인터페이스를 자동으로 생성하는 연구가 진행되고 있다[4-7]. 가전 기기에 다양한 기능을 추가하는 것은 제조사의 경쟁력을 높일 수는 있지만, 이러한 기기 제어를 위한 사용자 인터페이스 설계에 많은 비용이 필요하다[2]. 휴대 기기를 이용한 인터페이스 생성은 사용자의 이동에 따른 환경 변화를 인식하여 사용자와의 상호작용 과정을 돕고[8], 다른 플랫폼 상에서도 쉽게 사용될 수 있다[9-11]. 또한, 서로 다른 배경지식과 기기 사용 목적을 가지고 있는 사용자들의 요구를 수행할 수 있는 인터페이스 생성이 필요하다[12,13].

사용자는 가정환경의 각종 기기를 일일이 제어하기보다는 상위수준의 서비스를 제공받기 원한다. 예를 들어 TV, DVD, 5.1채널 스피커, 그리고 AV 리시버가 연결되어 있는 환경에서 사용자가 DVD 보기를 수행하기 위해서는 네 기기의 전원을 켜고, 기기들 간의 연결관계에 따라서 외부 입력을 설정해주어야 한다. 이러한 일련의 작업을 수행하려면, 사용자는 네 기기의 연결상태 및 기능과 각 기기의 리모컨 조작방법을 숙지해야 한다. 만일, 통합리모컨이 사용자가 원하는 작업에 따라 전원버튼을 한번만 누름으로써 이러한 사전 설정 작업이 자동으로 진행된다면, 사용자는 각 기기에 대해서 잘 알지 못하더라도 손쉽게 원하는 서비스를 제공 받을 수 있을 것이다. 본 논문에서는 다양한 기기들이 서로 연결되어 제공되는 서비스 작업단위를 태스크로 정의하고[14], 사용자가 원하는 서비스를 손쉽게 제공받을 수 있도록 태스크 단위 사용자 인터페이스의 생성을 위한 가전기기 마크업 언어(HAML; Home Appliance Markup Language)를 제안한다. HAML은 사용자의 이동성을 고려하여, 다른 환경에서도 동일한 인터페이스를 통해서 사용자가 원하는 서비스를 제공받을 수 있도록 하며, 초보자들도 추가적인 배경지식 없이 태스크 단위로 제공되는 인터페이스를 통해 간단한 조작만으로 원하는 서비스를 제공받을 수 있도록 한다.

2. 관련연구

2.1 사용자 인터페이스 기술언어

GUI(Graphic User Interface)의 사용이 많아짐에 따라서 인터페이스 개발에 많은 시간과 비용이 소요된다. 응용프로그램에서 인터페이스에 관련된 부분은 전체 코드의 47~60%를 차지하고[15], 전체 비용의 약 29%를 차지하고 있다[16]. 이러한 인터페이스 개발에 드는 많은 비용과 노력을 줄이기 위해서 사용자 인터페이스 기술언어(User Interface Description Language; UIDL)

가 등장하였다[17-19].

UIML[17]은 다양한 기기들의 UI를 하나의 범용적인 언어로 구축하여 그 개발기간을 단축하고, 유지 및 보수를 쉽게 하기 위한 목적으로 개발된 언어이다. UIML은 프로그램적인 접근보다는 선언적 접근을 사용하여 프로그래밍 언어를 배울 필요 없이 누구나 손쉽게 UI를 구현할 수 있으며, 다양한 기기에서 쉽게 변환이 가능하다. PIMA 시스템[18]은 UI 설계자가 다양한 기기에 맞는 UI를 설계할 필요 없이 추상화된 UI 모델을 설계하면, PIMA 시스템이 추상화된 모델을 실제 응용프로그램에 맞춰서 UI를 생성해준다.

2.2 통합리모컨

가정환경에서 다양한 기기들을 동시에 조작하도록 다수의 리모컨을 통합하려는 시도가 몇몇 업체들을 중심으로 진행되고 있다. 최근 TV 리모컨의 경우, 모드를 바꾸어 TV 뿐만 아니라 DVD 플레이어, 비디오, 오디오 등의 기기를 동시에 조작하도록 돕기도 하며, 소니는 각종 기기의 적외선 신호를 기억하여 하나의 리모컨으로 제어하는 통합 리모컨을 개발하기도 하였다. 뿐만 아니라 가정환경에서의 다양한 기기들에 접근하기 위해 마이크로소프트와 인텔을 중심으로 진행 중인 UPnP(Universal Plug and Play), 소니와 필립스 등이 제정한 HAVi(Home Audio Video interoperability), 썬 마이크로 시스템이 중심이 된 Jini, 그리고 에체렌의 제어 네트워크 기반기술인 Lonworks 등의 기술이 개발되었으며, ConvergeX 사의 경우, 이들 기반 기술을 이용하여 가정환경을 조작하는 통합 사용자 인터페이스를 구현하였고, HomeLogic 사는 이미 700개 이상의 가정환경 제어 시스템을 공급하였다. 하지만 이들 기술과 제품은 사용자가 가정기기의 각종 기능을 사용하도록 리모컨과 기기를 연결시켜 줄 뿐이어서 원하는 상위수준의 작업을 하기 위해서는 다수의 장치를 제어하기 위한 복잡한 조작이 필요하다.

기존의 다양한 기기의 리모컨을 하나로 통합하려는 시도는 사전에 정의된 환경에서만 적용이 가능하며, 사용자는 원하는 서비스를 제공받기 위해 여전히 많은 조작을 해야 한다는 단점이 있다. 이를 극복하기 위해 다중 기기를 효과적으로 제어하는 사용자 인터페이스에 관한 연구가 활발히 진행되고 있다.

Isbell 등은 사용자의 기기에 대한 사용 버튼을 분석하여 주요 태스크를 선정하고 사용자별로 개인화된 사용자 인터페이스를 생성하여 제공하였다[20]. 이는 다양한 기기를 스마트 폰이라는 하나의 매체를 통해 제어할 수 있다는 장점이 있지만, 여러 기기가 복합적으로 연동하여 제공하는 서비스나 기기 제어는 불가능하고 사전에 모든 기기의 인터페이스가 미리 설계되어야 한다.

통합리모컨 연구에서도 사용자 인터페이스 기술언어가 사용되었다. Nichols와 Myers는 가정과 사무실의 각종 기기를 XML형태로 기술하고 스마트폰으로 통합된 인터페이스를 제공하여 기기들을 제어하는 기술인 PUC(Personal Universal Controller)를 제안하였다[11]. 다양한 기기를 스마트폰이라는 하나의 매체를 통해 제어할 수 있다는 장점이 있지만, 여러 기기가 복합적으로 연동하여 제공하는 서비스나 기기 제어는 불가능하고 사전에 모든 기기의 인터페이스가 미리 설계되어야 한다. Nichols 그룹은 PUC 시스템을 기반으로 하여 사용자가 보다 편하게 복수의 기기를 제어하도록 음성과 영상 데이터 흐름 중심의 사용자 인터페이스를 자동으로 생성하는 HUDDLE 시스템을 개발하였다[21]. 사용자는 자신이 원하는 데이터의 소스를 선택하고, 각종 기기의 연결관계를 사용자가 명시하면 기기별로 사전에 설계된 인터페이스를 중심으로 연관된 기기들을 제어하기 위한 인터페이스가 자동으로 생성된다. HUDDLE은 하나의 화면에서 현재 작업에 필요한 다수의 기기에 대한 인터페이스가 제공되며, 사용자는 복수의 기기를 일일이 조작할 필요없이 상위 수준의 명령을 내리면 내부적으로 다수의 기기가 제어된다. 하지만 사용자가 원하는 작업을 수행하기 위해서는 기기간의 연결 관계를 숙지하고 기기 사이의 연결 관계를 명시해야 하는 단점이 있다.

3. HAML(Home Appliance Markup Language)

본 논문에서는 상위수준의 서비스 제공을 위해서, 사용자에게 제공되는 서비스 작업 단위를 태스크로 정의하고 태스크 수행을 위해서 동적으로 사용자 인터페이스

를 생성하기 위한 사용자 인터페이스 언어인 HAML을 제안한다. HAML은 XML기반의 마크업 언어로, 제작된 문서가 홈 네트워크 시스템에 존재하는 각종 기기에 대한 정보 및 상황의 표현 및 전송을 위해 사용되며, 사용자 단말에서 태스크 단위의 인터페이스 생성을 위해 사용된다. HAML의 설계 목적은 다음과 같다.

- 가전기기 제어를 위한 통합리모컨의 태스크 단위의 인터페이스를 제공한다.
- 사용자가 기기의 연결 상태 및 사용법을 숙지할 필요 없이 간단한 조작으로 기기를 제어할 수 있는 인터페이스를 제공한다.
- 태스크 수행에 꼭 필요한 버튼들만으로 인터페이스를 구성하여, 사용되지 않는 다수의 버튼에서 오는 사용자의 혼란을 줄인다.
- 다른 환경에서도 동일 태스크 수행을 위한 동일한 인터페이스를 제공한다.

통합리모컨이 태스크 기반 인터페이스를 구성하기 위해서는 사용자가 원하는 태스크의 정보와 주변 기기에 대한 정보가 필요하다. HAML은 통합리모컨이 필요한 정보를 얻을 수 있도록 정의된 언어로, 기기 기술언어와 태스크 기술언어, 그리고 동의어 기술언어로 구성된다. 본 논문에서는 홈씨어터 환경을 구성하는 가전기기들을 분석하고 이를 태스크 단위로 제어하는 마크업 언어를 제안한다.

3.1 기기 기술언어

기기 기술언어는 통합리모컨이 주변 기기 정보를 인식할 수 있도록 기기에 대한 정보를 기술하는 언어이다. 통합리모컨이 태스크 단위의 인터페이스를 구성하기 위

표 1 기기 리모컨의 기능 비교표

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|----------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [TV] 삼성 SPD-42D4S | O | O | O | O | O | O | O | O | O | O | O | X | X | X | X | X | X | X | O |
| [TV] 삼성 LN32R71BD | O | O | O | O | O | O | O | O | O | O | O | X | X | X | X | X | X | X | O |
| [TV] LG 23LC1R | O | O | O | O | O | O | O | O | O | O | O | X | X | X | X | X | X | X | O |
| [DVD] 삼성 SV-D596HD | O | X | X | X | X | X | O | O | O | X | X | O | O | O | O | O | O | O | X |
| [TV] 삼성 SV-C980HD | O | X | X | X | X | X | O | O | O | X | X | O | O | O | O | O | O | O | X |
| [TV] LG DV235 | O | O | X | X | X | X | O | O | O | X | X | O | O | O | O | O | O | X | X |
| [AVR] SOVICO AVR580 | O | O | O | O | O | X | O | O | O | O | O | X | X | X | X | X | X | X | O |
| [AVR] YAMAHA RX-V750 | O | O | O | O | O | X | O | O | O | O | O | X | X | X | X | X | X | X | O |

A: 전원 B: 숫자 C: 볼륨+/- D: 음소거 E: 채널+/- F: 이전채널 G: 메뉴
 H: 이동(상/하/좌/우) I: 선택 J: 입력선택 K: 취침예약 L: 재생/일시정지 M: 정지
 N: 팸리감기 O: 되감기 P: 한단계앞으로 Q: 한단계뒤로 R: 반복 S: 야간모드

해서는 주변 상황을 인식하고 기기들을 제어할 수 있어야 한다. 본 논문에서는 홈씨어터 환경을 구성하는 주요 기기인 TV, DVD, AV 리시버를 분석하여 기기의 내부 상태, 기기 간의 연결관계, 기기 조작방법을 위한 기기 기술언어를 설계한다.

표 1은 국내에서 생산된 대표적인 홈씨어터 기기의 리모컨 버튼을 보여준다. 가전기기는 리모컨의 각 버튼과 연관된 내부 상태 값을 가지고 있다. 기기의 내부 상태 값은 전원(on/off), 음향모드(표준/음악/영화 등) 등과 같이 열거형 변수로 표현되기도 하고, 채널, 음량, 명암, 밝기 등과 같이 숫자형 변수로 표현되기도 한다.

통합리모컨이 사용자가 원하는 태스크 환경에 맞는 인터페이스를 구축하기 위해서는 주변 기기에 대한 기본 정보와 각 기기의 리모컨이 가지고 있는 버튼에 대한 정보, 기기의 내부 상태, 버튼의 수행 결과 변하는 기기의 내부 상태, 그리고 다른 기기와의 연결상태 및 해당 연결을 위한 기기 세팅 방법이 필요하다.

그림 1은 기기 기술언어의 구조이다. 기기정보(dev_info)는 통합리모컨이 기기들을 구분하도록 하는 아이디

(dev_id)와 사용자에게 보여질 기기명(dev_name), 영상, 음성 등과 같이 각 기기가 생성하는 신호(dev_src)와 기기가 최종적으로 수신할 수 있는 신호(dev_des)가 명시된다. dev_src와 dev_des의 정보는 통합리모컨이 태스크 수행에 필요한 주변기기를 찾을 때 사용된다.

기기의 내부 상태를 표현하기 위한 기기변수(variable)는 숫자형 변수를 표현하기 위한 digit_variable과, 열거형 변수를 표현하기 위한 enum_variable로 구성된다. 변수는 mode 속성을 가지며, 숫자형 변수는 리모컨 버튼이 한번 눌러졌을 때 변화하는 내부 변수의 값을 표현하기 위해 step 속성을 갖는다. mode 속성은 태스크 수행을 위해 다른 기기들과 연관된 상황에서 기기의 연계 동작 방법을 명시한 것으로, parallel은 연결된 기기의 변수 값의 상대적 차이를 줄이는 방향으로 기기의 기능이 발현되며, harmonic의 경우는 연결된 기기의 변수 값이 모두 동일한 값을 갖도록 기능이 발현되고, src와 des의 경우는 각각 태스크 수행을 위한 src 기기 및 des 기기의 기능이 발현된다.

기기기능(function)은 기기 리모컨의 통신 방법을 나타내는 protocol과 리모컨의 각 버튼을 나타내는 dev_button들로 구성되며, dev_button은 버튼이 눌러졌을 때의 변수 값의 변화 특징에 따라서, 열거형 변수의 값을 한 단계씩 변화시키는 toggle 버튼과, 변수의 값을 특정 상태로 변화시키는 set 버튼, 그리고 숫자형 변수의 값을 증가 혹은 감소시키는 inc, dec 버튼으로 구성된다. 각 버튼은 통합리모컨이 해당 버튼을 사용할 때 기기에 보낼 신호 값을 포함하며, 버튼의 동작과 연관된 변수의 아이디를 포함한다.

기기연결(dev_link)은 다른 기기와의 연결 설정 상태 및 연결 설정 방법을 기술한다. src_concept과 dev_concept은 각각 어떤 종류의 소스가 어떤 기기와 연결되는지를 기술하고, state에서는 연결 설정을 위해 변수를 어떤 값으로 변경해야 하는지 기술한다. 통합리모컨은 기기연결 정보를 통해서 태스크 수행에 필요한 사전 세팅을 수행한다.

3.2 태스크 기술언어

가전기기 리모컨에 포함된 버튼은 약 50여 개에 달하지만 10~20 개의 버튼만이 사용된다[22]. 본 논문에서는 다수의 버튼에서 오는 사용자의 혼란을 피하기 위해서 태스크 제어에 꼭 필요한 버튼만을 모아 사용자 인터페이스를 구성한다.

통합리모컨이 태스크를 수행하기 위해서는 태스크 수행에 필요한 기기, 인터페이스 버튼, 그리고 버튼의 배치방법이 필요하다. 그림 2는 태스크 기술언어의 구조를 보여준다. 통합리모컨이 태스크를 수행하기 위해서는 먼저 태스크 수행에 필요한 환경이 갖추어졌는지 확인해

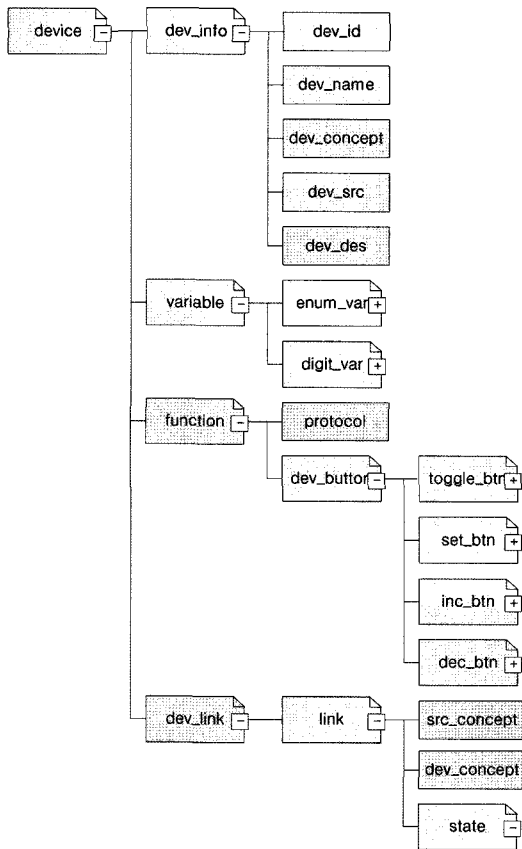


그림 1 기기 기술언어 구조

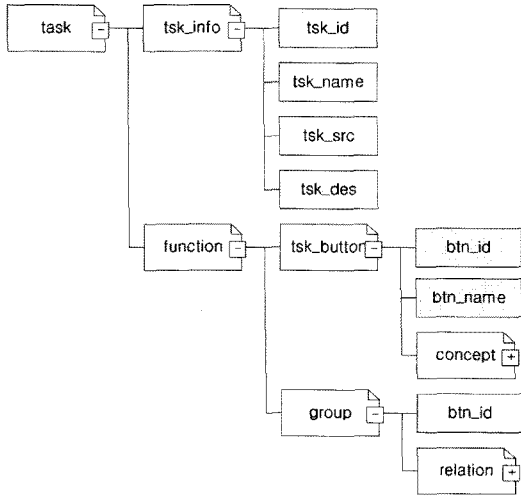


그림 2 태스크 기술언어 구조

야 한다. 태스크 수행을 위한 신호 생성 기기와 최종적으로 이 신호를 수신하는 기기가 무엇인지를 태스크 기술언어에 명시하고, 통합리모컨은 각 기기의 기기연결 정보를 바탕으로 어떤 기기와 연결되었는지를 따라가며 연결된 기기들을 찾는다. *tsk_src*와 *tsk_des*는 각각 태스크 수행을 위한 신호 생성 기기와 최종 신호 수신 기기를 명시한다. *tsk_src*를 통해서 태스크 수행의 소스 기기를 기술하거나 특정 기기의 신호를 기술한다. 비슷하게 *tsk_dev*를 통해서 최종적으로 처리할 신호나 신호 처리를 위한 기기를 기술한다.

*tsk_button*은 태스크가 선택되었을 때, 최종적으로 사용자 인터페이스에 나타날 버튼들로, 통합리모컨은 우선 태스크 수행에 필요한 기기들 중에서 *tsk_button*의 *concept* 정보가 일치하는 *dev_button*을 찾고, 해당 *dev_button*과 연결된 *variable*의 *mode* 값을 참조하여 적절한 *tsk_button*의 역할을 할당한다.

mode 값이 *harmonic*인 경우는 연결된 기기의 상태 값이 모두 동일한 상태를 유지하도록 하면서 변경시킨다. 만일 기기 상태가 동일하지 않을 경우에는, 가장 낮은 상태의 기기에 맞추어서 기기의 상태 값을 설정하고, 기기 상태가 동일한 경우는 모든 기기에 변경 신호를 전달한다.

*mode*의 값이 *parallel*인 경우에는 태스크 버튼과 연결된 기기 버튼의 연결 변수 값들의 상대적 위치를 고려하여 어느 기기의 상태 값을 변경시킬지 결정한다. 즉, 만일 DVD 음량(0~99)이 10으로 설정되어 있고, TV 음량(0~49)이 6으로 설정되어 있다면, DVD 음량의 상대 위치는 10/100이고, TV 음량의 상대 위치는 6/50이므로 ‘음량+’ 버튼이 눌러지면 상대 위치가 낮은 DVD

의 ‘음량+’ 신호가 전달되고, 반대로 ‘음량-’ 버튼이 눌러지면 상대 위치가 높은 TV ‘음량-’ 신호가 전달된다.

그리고 *src*의 경우는 해당 기능을 가진 기기 중에서 *tsk_src*에 가장 가까운 기기에게 변경 신호를 전달하며, 반대로 *des*의 경우에는 해당 기능을 가진 기기 중에서 *tsk_des*에 가장 가까운 기기에게 변경 신호를 전달한다.

*tsk_function*에서의 *group*은 유사한 속성을 갖는 버튼의 그룹을 의미하며, *relation*을 통해서 사용자 인터페이스 상에서 버튼의 배치 정보를 표현한다.

3.3 동의어 기술언어

동의어 기술언어는 기기 기술언어와 태스크 기술언어 간의 의미 정보 기반 연결을 위해서 사용된다. 기기 기술언어 및 태스크 기술언어는 그 작성자에 따라서 *id* 값이 다르게 설정될 수 있다. 만약 태스크 기술언어와 기기 기술언어, 그리고 이종의 기기 기술언어 간의 연결이 *id*를 이용하여 이루어 진다면, 사용자의 환경이 변화하는 경우 변화한 환경에 대해서 *id* 값을 일일이 수정해야 하는 번거로움이 존재한다. 본 논문에서는 이러한 추가적인 작업 없이 통합리모컨이 태스크 단위의 인터페이스를 생성하기 위해서 *id* 단위의 연결이 아닌, 의미 정보 단위의 연결을 제공한다.

각 기기에 대한 정보는 기기 제조사에서 생성해준다고 가정하면, A사에서는 전원 기능을 ‘power’로 기술하고, 동일 가전제품을 취급하는 B사에서는 ‘pw’로 기술하는 것처럼 같은 기능을 다르게 표기할 수 있는데, 의미 정보 기반 매핑을 통해 ‘power’와 ‘pw’가 동일한 기능을 가짐을 명시하여 해결할 수 있다.

4. 실험 및 결과

4.1 HAML을 이용한 통합리모컨의 구현

본 논문에서는 제안하는 방법의 유용성을 보이기 위해 그림 3과 같이 네 가지의 유사하지만 서로 다른 환경을 설정하고, 각 환경에서 통합리모컨이 HAML을 해석하여 어떻게 작동하는지 살펴본다.

그림 3(a)와 (b)는 TV와 DVD의 두 가지 가전기기로만 이루어진 환경인데 (a)의 경우는 DVD의 음성 및 영상 연결이 TV와 제대로 연결되어 있지만, (b)의 경우는 음성이 연결되어있지 않다. 그림 3(c)는 스피커와 AV 리시버가 추가로 주어져 있는 환경이고, (d)는 영상 수신 장치로 프로젝터가 추가되어 있다.

인터페이스 구성을 위한 기술언어를 다운로드한 후, 통합리모컨은 주변 기기를 인식하고 수행 가능한 태스크 목록을 작성한다. 여기에서는 태스크로 ‘TV 보기’와 ‘DVD 보기’를 예로 든다. ‘TV 보기’의 경우는 TV만이 있으면 수행 가능한 태스크이므로 특별한 태스크 검사가 필요하지 않다. 그러나 ‘DVD 보기’의 경우는 tsk_

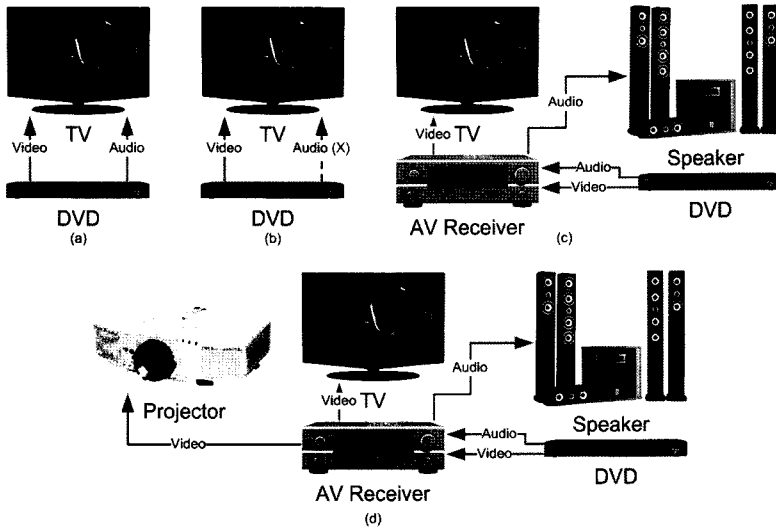


그림 3 홈 환경의 네 가지 예

| | |
|---|--|
| <pre> <device> <dev_info> <dev_id> TV </dev_id> <dev_name> TV </dev_name> <dev_concept> <concept> TV </concept> </dev_concept> <dev_src> <src_concept> <concept> 영상 </concept> </src_concept> <src_concept> <concept> 음성 </concept> </src_concept> </dev_src> <dev_des> <des_concept> <concept> 영상 </concept> </des_concept> <des_concept> <concept> 음성 </concept> </des_concept> </dev_des> </dev_info> <variable type = 'enum' mode = 'harmonic'> <var_id> power </var_id> <var_concept> <concept> 전원 </concept> </var_concept> <src_concept> <concept> 영상 </concept> </src_concept> <value> on <value_concept> <concept> 켜짐 </concept> </value_concept> </variable> <variable type = 'digit' mode = 'parallel' step = '1'> <var_id> vol </var_id> <var_concept> <concept> 소리 </concept> </var_concept> <src_concept> <concept> 음성 </concept> </src_concept> <min> 0 </min> <max> 100 </max> </variable> </pre> | <pre> <function> <protocol format = 'TC9012'> <button type = 'toggle' var_id = 'power'> <btn_id> d_power </btn_id> <btn_name> btn_name </btn_name> <btn_concept> <concept> 전원 </concept> </btn_concept> <ir_code delay = '1000'> 0110 1100 1001 0001 0000 0011 1111 0001 </ir_code> </button> <button type = 'inc' var_id = 'vol'> <btn_id> d_vol_up </btn_id> <btn_name> 소리 올리기 </btn_name> <btn_concept> <concept> 소리 </concept> <concept> 올리기 </concept> </btn_concept> <ir_code delay = '1000'> 0110 1100 1001 0001 0000 0010 1111 1101 </ir_code> </button> <button type = 'toggle' var_id = 'd_mute'> <btn_id> d_vol_off </btn_id> <btn_name> 음소거 </btn_name> <btn_concept> <concept> 음소거 </concept> </btn_concept> <ir_code delay = '1000'> 0110 1100 1001 0001 0000 0011 1111 0011 </ir_code> </button> </protocol> ... </function> <dev_link> <link type = 'input'> <src_concept> <concept> 영상 </concept> </src_concept> <dev_concept> <concept> DVD </concept> </dev_concept> <state> <variable var_id = 'ex_in' /> <value> 외부입력1 </value> </state> </link> ... </dev_link> </device> </pre> |
|---|--|

그림 4 'TV' 기기 기술언어의 예

| | |
|--|--|
| <pre> <task> <tsk_info> <tsk_id> watching_DVD </tsk_id> <tsk_name> watching DVD </tsk_name> <tsk_src> <dev_concept> <concept> DVD </concept> </dev_concept> </tsk_src> <tsk_des> <des_concept> <concept> Video </concept> </des_concept> </tsk_des> <tsk_des> <des_concept> <concept> Audio </concept> </des_concept> </tsk_des> <tsk_info> </function> </pre> | <pre> <button> <btn_id> power </btn_id> <btn_name> 전원 </btn_name> <btn_concept> <concept> 전원 </concept> </btn_concept> </button> ... <group> <grp_concept> <concept> 소리 </concept> </grp_concept> <button btn_id='d_vol_up'/> <button btn_id='d_vol_dn'/> <button btn_id='d_vol_off'/> <relation type='top2bottom'/> </group> ... </function> </task> </pre> |
|--|--|

그림 5 'DVD 보기' 태스크 기술언어의 예

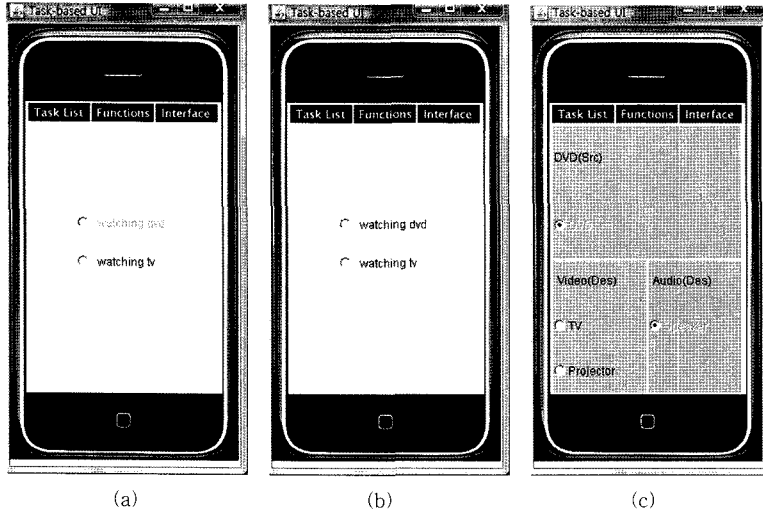


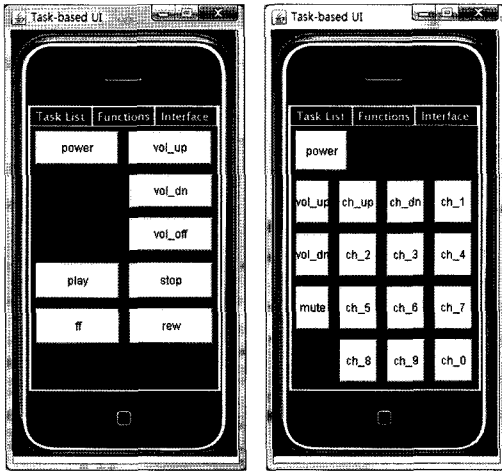
그림 6 태스크 선택 인터페이스: (a), (b) 태스크 선택 화면, (c) 기기 선택 화면

src로 DVD를 필요로 하며, tsk_des로 비디오 및 오디오 출력 장치가 필요하다(그림 5 참조). 그림 4와 5는 각각 기기 기술언어와 태스크 기술언어의 예로 그림 3(a)의 'TV'와 'DVD보기' 태스크를 보여준다.

그림 3(a), (c), (d)의 환경에서는 DVD의 음성/영상 연결이 TV, 프로젝터, 스피커 등의 출력장치와 올바르게 연결이 되어있는 반면에 (b)의 경우는 음성 연결이 되어 있지 않아서 'DVD 보기' 태스크의 수행이 불가능하다. 그림 6(a)는 그림 3(b)의 상황에서의, 그리고 (b)는 그림 3(a), (c), (d)의 상황에서의 태스크 선택 화면을 보여주고 있다.

그림 3의 네 가지 환경에서 'TV 보기' 태스크가 선택된 경우, 연관된 기기가 TV 뿐이므로 통합리모컨은 특별한 설정 없이 바로 'TV 보기'를 위한 컨트롤 인터페

이스를 생성하면 된다. 그러나 'DVD 보기'의 경우에는 연관된 기기들을 탐색하고 관련 기기간의 연결이 필요하다. 그림 3(a)의 경우에는 TV와 DVD 간의 연결이, (c)의 경우에는 TV - AV 리시버 - DVD 간에는 영상 연결, 스피커 - AV 리시버 - DVD 간에는 음성 연결이 필요하다. 3.2절에서 설명한 바와 같이 이러한 연결 설정은 통합리모컨에서 자동으로 설정된다. 하지만 그림 3(d)와 같이 영상 출력장치로 TV와 프로젝터를 모두 사용할 수 있는 경우에는 그림 6(c)와 같이 사용자의 피드백을 받아서 결정한다. 이때 tsk_src 기기와 tsk_des 중 음성 출력의 경우는 각각 DVD와 스피커로 결정되기 때문에 사용자가 선택할 필요가 없다. 이와 같이 제안하는 통합리모컨은 태스크 수행에 필요한 기기의 연결을 자동으로 수행해 주므로, 사용자 환경에서 기기들



(a) DVD 보기 (b) TV 보기
그림 7 사용자 인터페이스 예

이 제대로 연결되어 있다면, 사용자는 기기의 연결상태를 고려하지 않고, 원하는 태스크 선택만을 통해서 서비스를 제공받을 수 있다.

태스크 선택이 완료되면 통합리모컨은 태스크 수행을 위한 인터페이스를 생성한다. 태스크 기술언어에는 태스크 수행에 필요한 태스크 버튼에 대한 간단한 의미 정보만이 포함되어 있지만, 통합리모컨은 의미 정보 기반 매칭을 통해서 기기 기술언어에서 태스크 수행에 필요한 버튼들을 찾아 연결한다. 그림 7은 'DVD 보기'와 'TV 보기' 태스크 수행을 위한 사용자 인터페이스이다.

4.2 SUS 사용성 평가

제안하는 방법의 유용성을 보이기 위해 20~29세의 남녀 대학생과 대학원생 10명을 대상으로 모드를 바꾸어 다양한 기기를 제어할 수 있는 상용 통합리모컨 제품과 제안하는 방법을 비교하는 사용성 평가를 수행하였다. 사용성 평가는 표 2의 SUS(System Usability Scale)의 항목을 사용하여 진행하였다. 각 문항의 응답은 Likert 척도를 사용하여 강한 부정, 부정, 보통, 긍정, 강한 긍정의 1에서 5까지의 점수를 부여하도록 하였다. 홀수 문항은 평가 값에서 1을 빼고, 짝수 문항은 5에서 평가 값을 뺀 후 이를 모두 더하고, 여기에 2.5를 곱하여 0부터 100사이의 값을 갖는 점수로 환산한다.

피험자는 우선 TV, DVD가 갖춰진 환경에서 모드를 바꾸어서 다수의 기기를 제어할 수 있는 통합리모컨과 제안하는 통합리모컨을 이용하여 표 3과 같은 태스크를 수행한 후, AV 리시버가 추가된 환경에서 동일한 태스크를 수행하도록 하였다. 각 기기의 초기 상태는 표 4에서 보여주고, 표 5는 기기 환경에 따른 기기 연결 관계를 보여준다.

표 2 SUS 항목

| SUS | 문항 |
|--------|---------------------------------------|
| SUS-1 | 나는 이 시스템을 자주 사용할 것 같다. |
| SUS-2 | 시스템에 불필요하게 복잡한 부분이 있다. |
| SUS-3 | 시스템이 사용하기 쉽다고 생각한다. |
| SUS-4 | 이 시스템을 사용하기 위해서 전문가가 필요할 것 같다. |
| SUS-5 | 이 시스템은 다양한 기능이 조직적으로 잘 결합되어 있다. |
| SUS-6 | 이 시스템은 너무 불안정한 것 같다. |
| SUS-7 | 대부분의 사람이 이 시스템의 사용방법을 빨리 익힐 것이다. |
| SUS-8 | 이 시스템은 사용하기 귀찮은 부분이 있다. |
| SUS-9 | 나는 이 시스템을 사용했는데 자부심이 생긴다. |
| SUS-10 | 이 시스템을 계속 사용하기 위해서는 많은 것을 배워야 할 것 같다. |

표 3 사용자 태스크

| 태스크 | 내용 | |
|--------|---------|-----------------------|
| TV 보기 | 채널 변경 | TV의 채널을 11번으로 변경한다. |
| | 소리 시끄럽게 | 소리를 다소 시끄러운 정도로 설정한다. |
| | 소리 조용하게 | 소리가 작게 들릴 정도로 설정한다. |
| DVD 보기 | DVD 재생 | DVD를 재생한다. |
| | 소리 시끄럽게 | 소리를 다소 시끄러운 정도로 설정한다. |
| | 소리 조용하게 | 소리가 작게 들릴 정도로 설정한다. |

표 4 기기의 초기 상태

| 기기 | 상태 |
|-------|---|
| TV | 전원: off, 소리: 10, 채널: 35, 외부입력: TV |
| DVD | 전원: off, 상태: 정지 |
| AV리시버 | 전원: off, 소리: 10, 채널: 35, 외부입력 1, 외부출력 1 |

표 5 기기 환경에 따른 기기 연결상태

| 기기 환경 | 연결상태 |
|--------------|--|
| TV-DVD | TV 외부입력1에 DVD의 영상, 음성 연결 |
| TV-AC리시버-DVD | TV 외부입력1에 AV리시버의 영상, 음성 연결 AV리시버 외부출력2에 TV의 영상,음성 연결 AV리시버 외부입력2에 DVD의 영상, 음성 연결 |

그림 8은 실험 결과를 보여준다. 상황 1은 기기 환경이 TV와 DVD인 경우이고, 상황 2는 TV, AV리시버, 그리고 DVD를 이용한 경우이다. 실험 결과에서 보여주듯이 제안하는 통합리모컨은 기기의 주변 환경에 상관없이 약 20번의 리모컨 클릭 횟수를 보이지만, 일반 통합리모컨을 이용한 경우에는 기기의 개수가 많아짐에 따라서 설정해야 할 기기 상태가 많아지기 때문에, 평균 10회 이상의 설정이 추가적으로 필요함을 알 수 있다. 상황에 따른 버튼 클릭 횟수의 차이는 t-검정 결과, 상황 1, 2의 경우 모두 $p < 0.001$ (상황1: 2.91×10^{-6} , 상황 2: 7.29×10^{-8})로 통계적으로 유의한 결과이다.

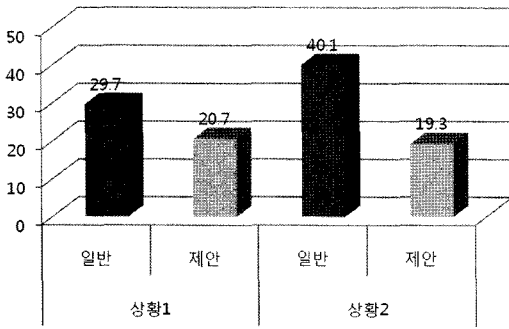


그림 8 상황에 따른 버튼 클릭 횟수

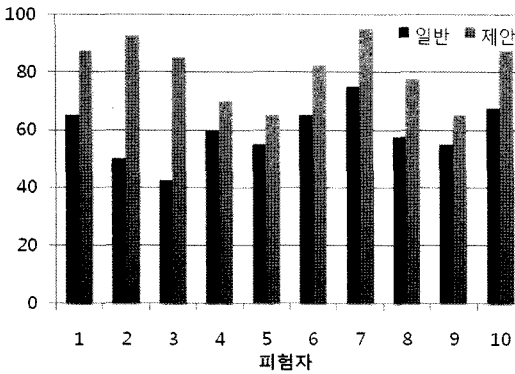


그림 9 SUS 사용성 평가 결과

그림 9는 일반 통합리모컨과 제안하는 통합리모컨의 SUS 사용성 평가 결과이다. 제안하는 통합리모컨의 사용성 평가 점수는 평균 80점으로 일반 통합리모컨의 58.6점보다 높은 사용성을 보이는 것을 확인하였다. 항목별로 보았을 때, 대부분의 경우 제안하는 통합리모컨이 일반 통합리모컨에 비해 높은 점수를 얻었지만, 문항 SUS-6에서는 제안하는 방법이 낮은 점수를 얻었다. 이는 하나의 태스크 버튼에 다양한 기기의 버튼이 매핑되는 경우, 각 기기의 기능이 수행되기까지 시간지연이 있어 피험자들이 시스템이 불안정하다고 느꼈기 때문이다. SUS 사용성 평가 결과의 차이는 t-검정 결과 $p < 0.001(1.56 \times 10^{-4})$ 로 통계적으로 유의한 결과이다.

5. 결론

본 논문에서는 홈 네트워크 환경에서 태스크 기반 사용자 인터페이스를 생성하기 위한 사용자 인터페이스 기술언어인 HAML을 제안하였다. HAML은 XML 기반의 인터페이스 기술언어로 홈 네트워크 환경을 표현하기 위해, 기기 기술언어, 태스크 기술언어, 동의어 기술언어로 구성되고, 통합리모컨은 HAML을 이용하여 사용자에게 태스크 단위의 사용자 인터페이스를 제공한

다. 또한, 통합리모컨은 태스크에 필요한 기기를 파악하고 필요한 사전 설정을 자동으로 함으로써 사용자는 기기의 연결 상태 및 사용법을 숙지할 필요없이 가전기기들을 조작할 수 있고, 사용자 인터페이스가 태스크 수행에 꼭 필요한 버튼들만 구성되어 간단한 조작만으로 기기를 제어할 수 있어 초보자들도 쉽게 사용할 수 있다. 그리고 동의어 기술언어를 통해서 기기 기술언어와 태스크 기술언어간의 의미 정보 기반 연결을 시도하므로, 통합리모컨은 기기 기술언어의 내용이 바뀌어도 사용자에게 동일한 서비스를 제공할 수 있다. 즉, 사용자는 자신의 홈 환경과 다른 기기 정보를 가진 환경에서도 동일한 인터페이스를 통해서 태스크를 수행할 수 있다.

실험에서 보여주듯이, 사용자는 TV보기, DVD보기 등과 같은 상위 수준의 작업단위인 태스크 선택을 통해서 간단하게 원하는 서비스를 수행할 수 있으며, 사용자의 특별한 설정없이 언제 어디서든 휴대 단말기를 통해 상위 수준의 기기제어 서비스를 제공받을 수 있다. 또한, 기기의 연결 관계에 대한 이해가 없어도 사용자가 손쉽게 원하는 서비스를 제공받을 수 있다.

향후 연구로는 태스크 버튼의 매크로 기능에서 발생하는 버튼간의 시간지연을 최소화 할 수 있는 방법이 필요하며, 다단계로 이루어진 각 기기의 메뉴에 대한 고려가 필요하다. 그리고 가정환경에서 보다 다양한 서비스와 관련된 태스크 기반 사용자 인터페이스를 제공하기 위해서 기기 명세와 태스크 명세를 확장해야 하고, 정보 소스나 의미 정보 사이의 관계를 보다 체계적으로 설계할 필요가 있으며, 기기나 태스크 명세의 설계를 위한 효과적인 설계틀의 개발이 필요하다. 또한 사용자에게 개인화된 사용자 인터페이스를 제공하도록 기기와 태스크를 분석하여 사용자 인터페이스를 생성할 때 개인의 성향이나 취향을 함께 고려하는 방법이 필요하다.

참고 문헌

- [1] G. Borriello and R. Want, "Embedded computation meets the world wide web," *Communications of the ACM*, vol.43, no.5, pp.59-66, 2000.
- [2] M.D. Brouwer-Janse et al., "Interfaces for consumer products: How to camouflage the computer," *Proc. ACM Conf. Human Factors in Computing Systems*, pp.287-290, 1992.
- [3] M. C. Filibeli, "Embedded web server-based home appliance networks," *Journal of Network and Computer Appliances*, vol.30, no.2, pp.499-514, 2007.
- [4] J. Nichols et al., "Demonstrating the viability of automatically generated user interfaces," *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp.1283-1292, 2007.
- [5] O. Omojokun et al., "Comparing end-user and

- intelligent remote control interface generation," *Personal and Ubiquitous Computing*, vol.10, no.2, pp.136-143, 2006.
- [6] C. Rich *et al.*, "DiamondHelp: A collaborative interface framework for networked home appliances," *25th IEEE Int. Conf. on Distributed Computing Systems*, pp.514-519, 2005.
- [7] J. Eisenstein, "Applying model-based techniques to the development of UIs for mobile computers," *Proc. of 6th Int. Conf. on Intelligent User Interfaces*, pp.69-76, 2001.
- [8] G. Calvary *et al.*, "Plasticity of user interfaces: a revised reference framework," *Proc. of 1st Int. Workshop on Task Models and Diagrams for User Interface Design*, pp.127-134, 2002.
- [9] K.Gajos *et al.*, "SUPPLE: Automatically generating user interfaces," *Proc. of 9th Int. Conf. on Intelligent User Interface*, pp.93-100, 2004.
- [10] G. Mori *et al.*, "Design and development of multi-device user interfaces through multiple logical descriptions," *IEEE Trans. on Software Engineering*, vol.30, no.8, pp.507-520, 2004.
- [11] J. Nichols *et al.*, "Generating remote control interfaces for complex appliances," *Proc. of 15th Annual ACM Symposium on User Interface Software and Technology*, pp.161-170, 2002.
- [12] K. Z. Gajos *et al.*, "Automatically generating custom user interfaces for users with physical disabilities," *Proc. of 8th Int. ACM SIGACCESS Conf. on Computers and Accessibility*, pp. 243-244, 2006.
- [13] J. Nichols *et al.*, "UNIFORM: Automatically generating consistent remote control user interfaces," *Proc. of SIGCHI Conf. on Human Factors in Computing Systems*, pp.611-620, 2006.
- [14] S. Lim and S. B. Cho, "A Task-based User Interface for Manipulating Multiply Connected Appliances," *Proc. of KCC*, vol.35, no.1(B), pp.171-175, 2008.
- [15] F. MacIntyre *et al.*, "Cost of user-friendly programming," *Journal of 4th Application and Research*, vol.6, no.2, pp.103-115, 1990.
- [16] D. Rosenberg, "Cost-benefit analysis for corporate user interfaces standards: What price to pay for a consistent look and feel," *Coordinating User Interfaces for Consistency*, New York Academic Press, pp.21-34, 1989.
- [17] M. Abrams *et al.*, "UIML: An appliance-independent XML user interface language," *Computer Networks*, vol.31, pp.1695-1708, 1999.
- [18] L.D. Bergman *et al.*, "Combining handcrafting and automatic generation of user interfaces for pervasive devices," *Proc. of Computer Aided Design of User Interfaces*, pp.155-166, 2002.
- [19] S. Berti *et al.*, "TERESA: A transformation-based environment for designing and developing multi-device interfaces," *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, pp. 793-794, 2004.
- [20] C. Isbell *et al.*, "From devices to tasks: Automatic task prediction for personalized appliance control," *Personal Ubiquitous Computing*, vol.8, no.3-4, pp. 146-153, 2004.
- [21] J. Nichols *et al.*, "Huddle: Automatically generating interfaces for systems of multiple connected appliances," *ACM Symposium on User Interface Software and Technology*, pp.279-288, 2006.
- [22] O. Omojokun and Jr. C. L. Isbell, "User modeling for personalized universal appliance interaction," *Proc. of 2003 Conf. on Diversity in Computing*, pp.65-68, 2003.



임 성 수

2004년 연세대학교 컴퓨터과학과(학사)
2006년 연세대학교 컴퓨터과학과(석사)
2006년~현재 연세대학교 컴퓨터과학과
박사과정. 관심분야는 인지 추론 및 학습, 지능형 에이전트



조 성 배

1988년 연세대학교 전산과학과(학사). 1990년 한국과학기술원 전산학과(석사). 1993년 한국과학기술원 전산학과(박사). 1993년~1995년 일본 ATR 인간정보통신연구소 객원 연구원. 1998년 호주 Univ. of New South Wales 초청연구원. 1995년~현재 연세대학교 컴퓨터과학과 정교수. 관심분야는 신경망, 패턴인식, 지능정보처리