

---

# 베이지안 네트워크와 행동 선택 네트워크를 이용한 유비쿼터스 홈에서의 상황 적응적 인터페이스 생성

## Context Adaptive User Interface Generation in Ubiquitous Home Using Bayesian Network and Behavior Selection Network

박한샘, Han-Saem Park\*, 송인지, In-Jee Song\*\*, 조성배, Sung-Bea Cho\*\*\*

---

**요약** 최근 가정환경의 홈 씨어터를 동작시키기 위해서는 TV, 오디오, DVD, Video, 셋탑박스 등 여러 장치를 동시에 조작해야 한다. 이 경우 사용자가 원하는 기능을 실행하기 위해서는 여러 기기에 해당하는 리모컨 버튼의 기능과 위치를 잘 알고 있어야 한다. 이러한 현실적인 문제로 인해 사용자들은 일반적으로 자신이 원하는 기능을 선택하는데 어려움을 겪는다. 더욱이 유비쿼터스 가정환경이 현실화 되어 사용자가 조작 가능한 장치들이 늘어나면, 사용자의 혼란은 가중될 것이다. 따라서 기능을 요약해서 사용자에게 제공하는 적응적 인터페이스가 필요하다. 또한, 유비쿼터스 환경에서는 조작하고자 하는 장치 뿐 아니라, 사용자 인터페이스가 표시되는 컨트롤러에도 다양한 모바일 또는 고정된 장치들이 사용되므로 각 장치의 능력이나 제약 조건에 맞게 사용자 인터페이스의 형태를 조절해 줄 필요가 있다. 제안하는 시스템에서는 상황에 따라 표현되는 기능과 형태가 변경되는 적응적 사용자 인터페이스를 구현하기 위해, 유비쿼터스 가정환경을 모델링하고, 모델링된 상황 및 장치 정보를 사용한다. 상황에 맞는 장치별 필요정도를 구하기 위해서는 베이지안 네트워크를 사용한다. 행동 선택 네트워크는 사용자의 상황과 예측된 장치별 필요도를 입력으로 사용해 장치별로 현 상황에 필요한 기능을 선택한다. 이렇게 선택된 기능들을 실제 사용자 인터페이스가 구현될 장치에 맞게 프레젠테이션 템플릿을 이용해 실제 사용자 인터페이스로 구성하여, 적응적 인터페이스를 구성한다. 실험을 위해서는 유비쿼터스 홈 시뮬레이션 환경을 구축하고, 해당 환경을 바탕으로 장치 사용기록을 시나리오를 바탕으로 생성하였다. 생성된 시나리오를 바탕으로 장치별 필요도 추론결과를 평가하여 베이지안 네트워크가 효과적으로 사용자의 요구를 예측함을 확인하였다. 마지막으로, 14명의 사용자들에게 평가된 10개의 태스크에 대해 기존의 고정된 홈 UI와 제안하는 적응적 홈 UI를 비교해 본 결과, 생성된 적응적 홈 UI가 일반적인 태스크를 고정적 홈 UI에 비해 효과적으로 처리할 수 있음을 확인하였다.

**Abstract** Recently, we should control various devices such as TV, audio, DVD player, video player, and set-top box simultaneously to manipulate home theater system. To execute the function the user want in this situation, user should know functions and positions of the buttons in several remote controllers. Normally, people feel difficult due to these realistic problems. Besides, the number of the devices that we can control shall increase, and people will confuse more if the ubiquitous home environment is realized. Therefore, user adaptive interface that provides the summarized functions is required. Moreover there can be a lot of mobile and stationary controller devices in ubiquitous computing environment, so user interface should be adaptive in selecting the functions that user wants and in adjusting the features of UI to fit in specific controller. To implement the user and controller adaptive interface, we modeled the ubiquitous home environment and used modeled context and device information. We have used Bayesian network to get the degree of necessity in each situation. Behavior selection network uses predicted user situation and the degree of necessity, and it selects necessary functions in current situation. Selected functions are used to construct adaptive interface for each controller using presentation template. For experiments, we have implemented ubiquitous home environment and generated controller usage log in this environment. We have confirmed the BN predicted user requirements effectively as evaluating the inferred results of controller necessity based on generated scenario. Finally, comparing the adaptive home UI with the fixed one to 14 subjects, we confirmed that the generated adaptive UI was more useful for general tasks than fixed UI.

**핵심어:** Adaptive user interface, ubiquitous home, Bayesian network, behavior selection network

본 연구는 삼성전자(주)종합기술원에 의해 지원되었음

\*주저자 : 연세대학교 컴퓨터과학과 박사과정 e-mail: sammy@sclab.yonsei.ac.kr

\*\*공동저자 : 연세대학교 컴퓨터과학과 석사 e-mail: schunya@sclab.yonsei.ac.kr

\*\*\*교신저자 : 연세대학교 컴퓨터과학과 교수; e-mail: sbcho@cs.yonsei.ac.kr

## 1. 서론

가정에서 가전제품을 조작하기 위해서는 제품 구매시에 주어지는 리모컨을 사용한다. 구입하는 전자기기가 늘어나게 되면, 사용자가 사용해야 할 리모컨의 수는 그에 따라 늘어나게 된다. 예를 들어, 최근의 홈 씨어터 시스템을 사용하기 위해서는 케이블 또는 위성 TV 셋탑박스, DVD 플레이어, TV, 오디오, 비디오, DivX 플레이어와 같이 다양한 기기를 동시에 제어해야 한다. 다른 기기를 위한 리모컨은 유사해 보이지만, 기기나 제조사에 따라 디자인이 다르다. 따라서 많은 장치를 위한 리모컨을 모두 익숙하게 사용하는 것은 생각보다 어려운 일이다[1]. 기기를 위해 제공되는 리모컨의 기능은 굉장히 다양하지만 실제로 사용되는 버튼은 전체의 1/3정도밖에 되지 않는다. 향후 유비쿼터스 컴퓨팅 환경이 가정환경에 구현되게 되면 이 외에 일반적인 기기들도 리모컨으로 조작을 하게 될 것이다. 따라서 여러 기기를 위한 다양한 기능들 가운데 사용자에게 필요한 기능을 제공하는 상황 적응적인 사용자 인터페이스가 요구된다.

제안하는 시스템에서는 상황에 따라 표현되는 기능과 인터페이스가 바뀌는 적응적 사용자 인터페이스를 구현하기 위해, 먼저 유비쿼터스 가정환경을 모델링하여, 모델링된 상황 및 기기 정보를 활용하였다. 사용자의 입력과 센서로부터 얻을 수 있는 상황 이외에 베이지안 네트워크를 사용하여 상황에 맞는 기기별 필요 정도를 추론하고자 하였다. 또한 사용자의 상황과 예측된 기기별 필요도를 행동 선택 네트워크의 입력으로 사용하여, 각 기기별로 현 상황에 필요한 기능을 선택하였다. 이렇게 선택된 기능을 사용자 인터페이스가 구성될 장치에 맞는 프리젠테이션 템플릿을 이용하여 구성하여 적응적 인터페이스를 구현하였다. 베이지안 네트워크를 사용한 사용자 모델링은 사용자의 사전 지식을 활용하기 쉬워 사용자 의도 예측과 같은 학습이 어려운 문제를 모델링하고자 할 때, 좋은 성능을 얻을 수 있다[2]. 또한 행동 선택 네트워크를 변형하여, 추상적 사용자 인터페이스가 사용자의 입력에 따라 지속적으로 변화할 수 있도록 효과적으로 모델링하였다.

## 2. 관련연구

### 2.1 유비쿼터스 홈 사용자 인터페이스

최근 다양한 기기들을 동시에 조작 할 수 있는 통합 리모컨이나 PDA용 프로그램 등이 도입되고 있다. 이를 사용하기 위해서는 기존의 기기별 리모컨에서 송출하는 적외선 신호를 기억해 사용자가 원하는 버튼에 설정해 주어야 하며, 이를 위한 프로그래밍 작업을 거쳐야 한다. 이와 같은 부가적인 과정 없이 기기 조작이 가능하도록, 가정환경의 모든 장치들을 미리 연결하여 어디서나 기기에 접근이 가능하도록 하는 홈 네트워크 기술이 연구되고 있다. 홈 네트

워킹을 위한 대표적인 기술은 마이크로소프트와 인텔을 중심으로 진행중인 UPnP (Universal Plug and Play)<sup>1)</sup>, 소니와 필립스 등이 제정한 HAVi (Home Audio Video interoperability)<sup>2)</sup>, 선마이크로시스템즈가 중심이 되어 진행중인 Jini<sup>3)</sup> 등이 있다.

하지만 홈 네트워킹 기술은 기기를 서로 연결시켜 줄 뿐이어서, 실제로 기기를 동작시키기 위해서는 프로그래머가 기기 스펙과 기능을 미리 알고 그에 맞게 사용자 인터페이스를 디자인해야 한다. 따라서 조작하고자 하는 기기가 늘어날 때마다 해당 기기를 위한 인터페이스를 추가해 주어야 하고, 컨트롤러가 변경될 때마다 다른 인터페이스가 구현되어야 한다. 이러한 문제점을 극복하기 위해 각 기기와 컨트롤러의 스펙과 기능을 표현하는 표준을 정하기 위한 노력이 진행되고 있다[3].

홈 네트워크 기술과 표준화된 기기 스펙을 활용하면 공개적인 기기 스펙으로 정의된 모든 기기에 대한 조작이 가능하다. 하지만 여전히 사용자 인터페이스는 해당 컨트롤러에 의존적으로 설계되어야 하기 때문에, 어떠한 컨트롤러에서도 실행되는 가정환경 기기 조작용 사용자 인터페이스를 구현하기 위해서는 사용자 인터페이스를 사용자가 원하는 컨트롤러에 맞게 변형해 주는 기술이 필요하다.

### 2.2 개인용 통합 컨트롤러

기기 제어를 위한 사용자 인터페이스를 PDA나 스마트폰과 같은 모바일 디바이스에 자동으로 생성해 주는 개인용 통합 컨트롤러(PUC, Personal Universal Controller)에 대한 연구도 최근 진행되고 있다. CMU의 J. Nichols와 B. A. Myers는 가정 혹은 사무실의 가전 기기제어를 위한 컨트롤러 인터페이스를 스마트 폰에 자동으로 생성하여, 소형의 스크린에 최적화 될 수 있도록 계층적 메뉴 네비게이션이 가능한 시스템을 개발하였고[4], 사용자가 원하는 태스크를 중심으로 모바일 디바이스에 해당 태스크를 위한 사용자 인터페이스를 자동 생성해 주는 시스템인 HUDDLE을 개발하였다[1]. 또한 사용자 스테디를 통해 자동 생성된 인터페이스가 일반 기기 제어를 위한 인터페이스와 비교해 사용성(정해진 태스크에 대한 수행 속도)이 더 좋다는 것과 새로운 인터페이스가 생성될 때, 이전 시스템과의 일관성(consistency)이 더 낫다는 점을 입증하였다[5].

이러한 연구들은 사용자에게 편리한 PUC를 생성하여 제공하지만, 사용자의 현재 상황을 고려해 주지는 못한다. 본 논문에서는 센서로부터 입력받은 사용자의 현재 상황 정보를 바탕으로 사용자에게 필요한 기기 및 기능을 추론해서 제어를 위한 사용자 인터페이스에 반영한다.

1) <http://www.upnp.org/>

2) <http://www.havi.org/>

3) <http://www.jini.org/>

### 2.3 상황 인지를 위한 베이지안 네트워크

상황(context)은 여러 가지 의미로 사용될 수 있는데, Dey는 “사용자와 애플리케이션의 상호작용과 관련이 있는 사람, 장소, 물체의 상황을 특징짓기 위해 사용될 수 있는 모든 정보” 라고 상황을 정의하였다[6]. 일반적으로 상황 정보는 사용자의 서비스에 대한 선호도에 큰 영향을 끼치며, 가정환경에서의 기기 조작에도 마찬가지로이다. 현재 사용자가 처한 상황에 따라, 사용자가 조작하고자 하는 기기가 달라질 수 있기 때문이다.

상황 인지를 위해서 사용되는 다양한 모델 중 하나인 베이지안 네트워크(Bayesian network)는 불확실하고 불완전한 정보를 이용하여 상황을 추론하는 안정적인 모델이다 [2]. 데이터에 의한 학습과 전문가에 의한 모듈의 설계가 모두 가능하며, 이러한 장점을 바탕으로 모바일 컨텍스트의 분류나 추론을 위해 많이 사용되어 왔다. VTT의 Korpipaa 연구팀은 모바일 디바이스 사용자의 상황을 학습, 분류하기 위해 나이브 베이즈(naive Bayes) 모델을 사용하였다[7]. 또한 마이크로소프트 연구소(MS Research)의 E. Horvitz 연구팀은 불확실한 환경에서 베이지안 네트워크를 사용하여 사용자가 현재 어디에 집중하고 있는지 추론하는 시스템을 제안하였다[8].

### 2.4 행동 선택 네트워크

P. Maes가 로봇 에이전트의 제어를 위해 제안한 행동 선택 네트워크의 구조는 그림 1과 같이 표현된다[9]. 네트워크는 여러 기본 행동으로 이루어져 있으며, 각 행동을 구성하는 선행조건(precondition), 추가조건, 삭제조건, 활성화도, 수행코드를 가지고 있고, 각각의 내부 연결과 외부 연결을 갖는다. 내부 연결은 선행조건 연결(predecessor link)과 후계자 연결(successor link), 충돌자 연결(conflict link)이 있으며, 외부 연결은 센서와 목적으로의 연결이 있다.

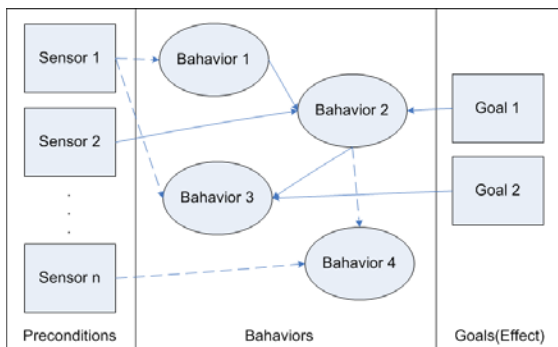


그림 1. 행동 선택 네트워크의 구조

행동 선택 네트워크를 구성하기 위해서는 먼저 기본 행동을 정의하고, 센서로부터 얻을 수 있는 정보와 목적을 연결하는 선행 조건 연결, 후계자 연결, 충돌자 연결을 정의

한다. 각 정보를 정의는 다음과 같이 이루어진다.

- 1) 행동 선택 네트워크를 구성하는 각 노드 정보
  - 선행조건: 노드가 참이 되기 위한 조건
  - 추가조건: 노드가 참이 되는 조건
  - 삭제조건: 노드가 거짓이 되는 조건
  - 활성화도: 행동 선택을 위한 가중치
  - 수행코드: 노드를 실행시키기 위한 코드
- 2) 행동 선택 네트워크의 내부 연결
  - 선행조건 연결: A행동의 추가조건에 있고 B행동의 추가조건에 있을 경우
  - 후계자 연결: A행동의 선행조건에 있고, B행동의 추가조건에 있을 경우
  - 충돌자 연결: 조건이 거짓이고 A행동의 선행조건이면서, B행동의 삭제조건에 있을 경우
- 3) 행동 선택 네트워크의 외부 연결
  - 센서: 센서 정보를 이용해 정의
  - 목적: 에이전트의 행동의 목적을 이용해 정의

### 3. 유비쿼터스 홈을 위한 상황 적응적 사용자 인터페이스

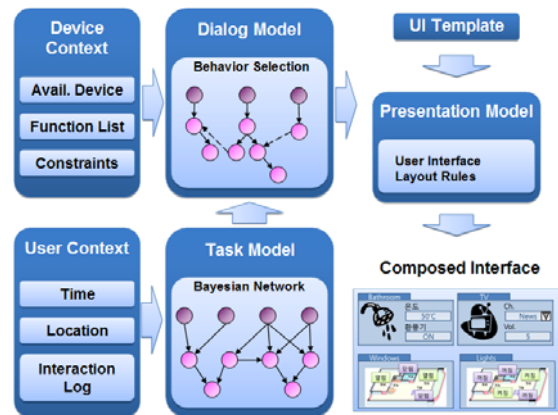


그림 2. 상황 적응적 사용자 인터페이스 구성하기 위한 방법

본 논문에서는 유비쿼터스 홈의 다양한 기기를 효과적으로 제어할 수 있도록 상황 적응적 사용자 인터페이스를 구성하기 위해 그림 2와 같은 방법을 사용한다. 먼저 사용자와 가정환경의 상황 정보를 사용자가 필요한 기기를 예측하는 베이지안 네트워크의 입력으로 사용하여, 사용자가 필요할 것으로 예상되는 기기 목록과 기기별 필요정도를 구한다. 이 예측 결과를 기기 및 컨트롤러 명세와 함께 사용하여 기능 선택을 위해 행동 선택 네트워크를 구성, 기기별로 사용할 기능 목록을 선택한다. 선택된 기능 목록을 사용자 인터페이스로 표현하기 위해, UI 템플릿을 사용하여 기능별로 적합한 형태로 바꾼 뒤 전체 레이아웃을 조정하여 화면에 표시한다.

### 3.1 가정환경 모델링

유비쿼터스 가정환경을 위한 적응적 사용자 인터페이스 구성을 위해 다음과 같이 가정환경을 모델링하였다.

- $E = \{Location\_List, Device\_List, Sensor\_List\}$  : 가정 환경, 하나의 집을 표현하며, 위치 목록, 장치 목록, 센서 목록으로 구성됨
- $Location\_List = \{Location1, Location2, \dots, LocationN\}$
- $Device\_List = \{Device1, Device2, \dots, DeviceN\}$
- $Sensor\_List = \{Sensor1, Sensor2, \dots, SensorN\}$
- $Location = \{Location\_Name, Location\_ID, Neighbors\}$
- $Device = \{Device\_Name, Device\_ID, Location\_ID, Function\_List\}$
- $Function\_List = \{Function1, Function2, \dots, FunctionN\}$
- $Function = \{Function\_Name, Function\_ID, Function\_Type, Function\_Value, Add\_List, Delete\_List\}$
- $Sensor = \{Sensor\_Name, Sensor\_ID, Sensor\_Type, Sensor\_Value\}$

유비쿼터스 가정환경은 크게 TV, 오디오 등의 가전기와 창문, 커튼, 보일러 등의 설비를 포함한 기기 목록과 각 방과 같은 가정환경내의 공간 정보를 표현하는 위치, 그리고 가정환경과 사용자의 상태를 감지하는 센서로 구성된다. 기기 정보는 해당 기기의 이름, 위치, 기기가 갖고 있는 기능 목록으로 이루어진다. 기능은 각 기기의 기능과 함께 제약 조건을 표현한다. TV의 경우 전원 조작, 소리 조작 등의 능력과 화면 크기 등의 제약 조건이 함께 기능 목록에 포함된다. 그리고 기능은 후에 기능 선택 시에 사용되는 행동 선택 네트워크 구성을 위해서, Add\_List와 Delete\_List를 갖는다. Add\_List는 해당 기능이 표시 되었을 때 함께 표시되면 좋을 기능들의 목록을 갖고, Delete\_List는 해당 기능이 표시되었을 때 함께 표시되지 말아야 할 기능들의 목록을 갖는다. 위치 정보는 각 장소의 이름으로 주로 방단위로 표현되었다. 마지막으로 가정환경의 상황 인지를 위해 센서가 표현된다.

### 3.2 베이زي안 네트워크를 이용한 사용자 필요장치 예측

가정환경에서 특정 상황에 사용자가 필요할 것으로 예상되는 장치 목록을 추론하기 위해 베이زي안 네트워크를 사용하였다. 베이زي안 네트워크는 초기 학습 데이터가 없는 상태에서도 전문가의 지식에 의존하여 네트워크를 설계할 수 있으므로[2], 적응형 인터페이스가 가정에 처음 설치되더라도 직접 설계한 베이زي안 네트워크를 사용해 홈 환경의 불확실성을 처리하여 안정적인 성능을 보일 수 있다. 그림 3은 상황 정보와 기존 기기 사용기록을 바탕으로 설계된 베이زي안 네트워크의 예이다.

사용자의 기기 사용기록이 어느 정도 모이면, 이 기록을 바탕으로 베이زي안 네트워크를 학습하는 것도 가능하다. 이 경우 각 사용자의 장치 사용 패턴을 학습한 개인화 된 베

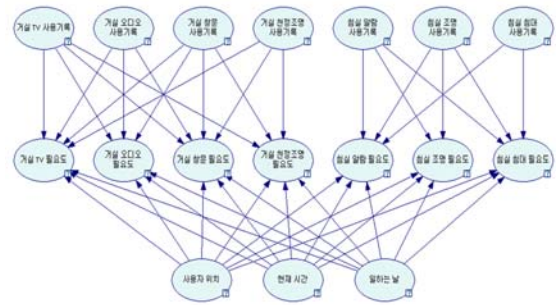


그림 3. 장치별 사용자 필요도를 예측하기 위한 베이زي안 네트워크의 예

이지안 네트워크를 생성할 수 있다. 본 논문에서는 구조 학습을 위해 K2 알고리즘을 사용하였다[10].

특정 기기의 필요도를 계산하기 위해서는 기본적인 상황 정보인 사용자의 위치나 현재 시간대, 공휴일인지 여부 등이 증거 변수로 사용된다. 상황 정보와 함께 각 기기와 연관된 기기의 사용기록이 함께 증거 변수로 사용된다. 주로 해당 기기의 과거 사용기록과 같은 방에 있거나 같은 종류의 기기 사용기록이 함께 증거 변수로 사용된다. 모든 증거 변수가 설정되면 BN 추론을 수행하여 기기의 필요도를 예측하게 된다.

### 3.3 행동 선택 네트워크를 이용한 필요기능 선택

상황 인식 베이زي안 네트워크를 사용해 장치별 필요도가 구해지면, 3.1에서 정의된 것과 같은 가정환경의 장치 목록  $F$  및 기능 목록  $D$ 를 사용해 행동 선택 네트워크가 설계된다. 현재 가정환경의 상태를 반영해 베이زي안 네트워크로부터 얻은 각 장치별 필요도를 환경 노드 집합  $E$ 라고 하고, TV 전원의 상태와 같은 가정환경 장치가 갖는 기능들의 상태를 기능 노드 집합  $B$ 라고 하면  $E$ 와  $B$ 는  $F$ 와  $D$ 를 사용해 식 (1), (2)와 같이 구성된다.

$$E = \{e_i | Exuete(e_i) \in D\} \quad (1)$$

$$B = \{b_i | Exuete(b_i) \in F\} \quad (2)$$

이와 같이 노드를 구성한 후에, 각 노드 사이에 선행조건 연결, 후계자 연결, 억제 연결 이렇게 세 종류의 연결이 생성된다. 선행조건 연결은 식 (3)과 같이 정의된다. 선행조건 연결은 두 기능 노드 사이에서 또는 기능 노드와 환경 노드 사이에서 생성 가능한 연결로, 두 노드가 한 기기에 속해 있고, 두 노드 사이에 긍정적인 연관 관계가 존재할 때 생성된다. 여기서 긍정적인 연관 관계는 특정 기능을 실행 했을 때, 그 다음에 실행될 가능성이 있는 기능들 사이에서 생긴다. 이 선행조건 연결은 한 장치에 존재하는 기능들을 계층적으로 구성하기 위해 사용된다. 서로 다른 기기에 존재하는 기능들 사이에 연관관계를 표현하기 위해 활성화 연결과 억제 연결을 사용하였다. 활성화 연결은 식 4를



사용해 결정된다. 활성화 연결의 경우 선행조건 연결의 경우가 유사하지만, 다른 것은 서로 다른 기기에 존재하는 기능들 간에 생성된다는 점과 기능 노드 사이에서만 생성된다는 점이다. 그리고 억제 연결은 식 (5)를 통해 결정되는데, 활성화 연결과 반대로 두 기능 사이에 부정적인 연관관계가 있을 때 생성된다.

$$Precondition(n_p, n_s) = \begin{cases} 1 & \text{if } \left\{ \begin{array}{l} (n_p \subset B \vee n_p \subset E) \wedge n_s \subset B \\ Device(n_p) = Device(n_s) \\ Relation(Execute(n_p), Execute(n_s)) \end{array} \right\} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$Successor(n_p, n_s) = \begin{cases} 1 & \text{if } \left\{ \begin{array}{l} n_p \subset B \wedge n_s \subset B \\ Device(n_p) = Device(n_s) \\ Relation(Execute(n_p), Execute(n_s)) \end{array} \right\} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$Conflictor(n_p, n_s) = \begin{cases} 1 & \text{if } \left\{ \begin{array}{l} n_p \subset B \wedge n_s \subset B \\ Device(n_p) = Device(n_s) \\ Confliction(Execute(n_p), Execute(n_s)) \end{array} \right\} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

이렇게 구성된 행동 선택 네트워크는 그림 11에서 보듯이 기본적으로는 각 기기 노드를 루트로 하는 트리 형태에 기능 간 연관관계에 따라 연결들이 추가된 형태를 갖는다. 이 행동 선택 네트워크는 각 상태에서 조건에 따라 활성화되는 활성화 함수는  $F: E \times B \rightarrow [0..1]$ 를 통해 평가한다. 이전 단계에서 상황인식 베이지안 네트워크를 사용해 예측한 각 장치별 필요도를 사용해 환경 노드 집합  $E$ 를 활성화한다. 그 다음 2.4에 제시된 행동 선택 네트워크 알고리즘을 통해 네트워크를 평가하면, 각 기기별로 전달된 활성화도가 활성화 또는 억제 연결을 따라 기능 노드들을 활성화한다. 이 활성화 과정에서는 기존의 행동 선택 네트워크와 같은 방법을 사용한다. 각 기능 노드  $b_i$ 의 활성화도  $a_i$ 를 구한 후, 현재 상태가  $t$ 인 행동선택 네트워크에서 활성화 기능 노드  $b_i$ 를 선택하기 위해서는 식 6이 사용된다. 기존의 행동 선택 네트워크와 달리 제안하는 방법에서는 여러 개의 기능이 동시에 사용자 인터페이스에 표시되므로, 단 하나의 기능이 아닌 여러 개의 기능들이 동시에 선택 될 수 있게 하였다. 기능 노드들이 이와 같이 선택되면 다음 사용자 인터페이스 구성 단계에서 사용된다.

$$b_i(t) = \begin{cases} 1 & \text{if } \left\{ \begin{array}{l} a_i(t) \geq \theta \\ executable(b_i, t) = 1 \end{array} \right\} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

이렇게 한번 구성된 행동 선택 네트워크는 사용자가 필요로 하는 기기 목록이 크게 바뀌거나, 또는 사용자의 직간접적인 요청에 의해서 다시 평가 되는데, 그림 4의 예와 같이 사용자가 TV를 시청하는 경우, TV의 필요도가 커지므로 행동 선택 네트워크의 TV 기기 노드의 활성화도가 다시 설정된다. 이 행동 선택 네트워크를 다시 평가하면, TV 관련 기능 노드들이 추가로 선택되어 사용자 인터페이스 상에서 TV 관련 기능이 더 많이 표시될 것이다.

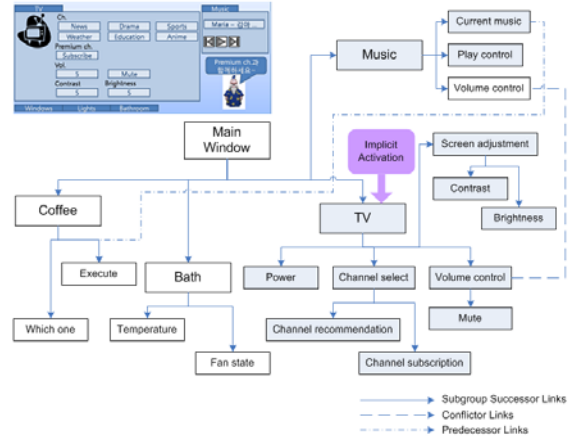


그림 4. 사용자가 TV를 조작한 후의 행동 선택 네트워크상의 활성화 변화와 이로부터 구성된 사용자 인터페이스

## 4. 실험 및 결과

### 4.1 실험 환경

유비쿼터스 가정환경을 시뮬레이션 하기 위해 그림 5와 같이 가정환경을 모델링하였다. 거실, 주방, 침실, 서재, 화장실, 5개의 방으로 구성되며, 각 방엔 방별로 적절한 기기가 존재한다. 기기 목록 및 기능 목록은 XML 형태로 정의하였고 BN과 행동 선택 네트워크는 스크립트 언어인 Python을 이용해 구현하였다.

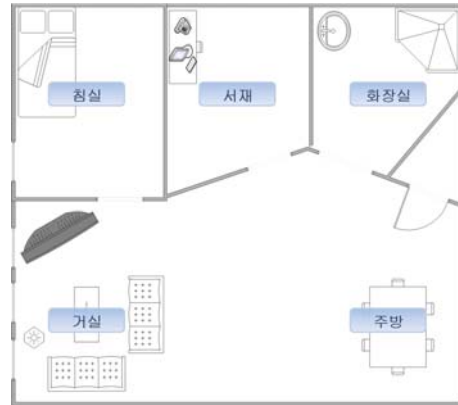


그림 5. 가정환경 평면도

### 4.2 적응적 사용자 인터페이스의 사용성 평가

제안하는 시스템의 사용성을 검증하기 위해 가정에서 자주 발생할 만한 10개의 상황을 가정하고, 생성된 적응적 사용자 인터페이스의 사용성을 GOMS[11]를 이용해 평가하였다. 10개의 상황은 4.1에서 언급된 홈 환경에서 발생 가능한 것으로 선정되었다. 예를 들어 상황 1은 “오디오를 들으며 아침 식사하기”이며 이를 위해 사용자가 수행해야 하는 세부 과제의 수행 순서는 다음과 같다: 주방 천장 조명을 켜다 → 주방의 커피 머신을 동작시킨다 → 거실 오디오를 켜다 → 거실 오디오 채널을 설정한다 → 거실 오디오 볼

룸을 조정한다 → (식사) → 주방 천장 조명을 끈다.



그림 6. 고정 UI만을 갖는 사용자 인터페이스



그림 7. 고정 UI와 적응적 UI를 모두 갖는 사용자 인터페이스

이러한 10가지 상황을 바탕으로 그림 6과 7의 사용자 인터페이스를 GOMS를 통해 평가하였다. 그림 6은 장소 및 기기 랩을 사용해서, 각 기기를 조작 할 수 있도록 구성된 고정된 홈 사용자 인터페이스이다. 현재 홈 네트워크 환경에서 많이 사용되고 있는 인터페이스를 바탕으로 설계되었다. 비교를 위해 그림 7과 같은 사용자 인터페이스를 구현하였다. 이 인터페이스는 그림 6의 인터페이스의 기능에 추가적으로 상황 적응적으로 구성되는 인터페이스를 추가한 것이다. 사용자의 장치 조작 및 상황에 따라 오른쪽에 위치한 인터페이스가 동적으로 변화한다.

그림 6과 7의 두 인터페이스를 사용하여 10가지 상황에 따라 주어진 세부과제를 수행하도록 하고 전체 과제 수행 시간을 측정하였고, 그 결과는 그림 8과 같다. 적응적 UI를 함께 사용하였을 때, 소요시간의 감소 비율은 38%로 상황에 따른 과제 처리 속도가 향상된 것을 확인 할 수 있었다.

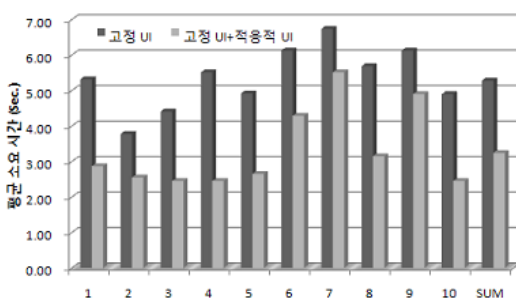


그림 8. 상황별 세부과제 수행을 위한 소요시간(고정UI vs. 고정 UI+적응적UI)

## 5. 결론

본 논문은 유비쿼터스 가정환경을 위한 적응적 사용자 인터페이스 생성 방법을 제안한다. 유비쿼터스 가정환경에서는 제어하고자 하는 기기뿐 아니라 제어하기 위한 컨트롤러도 다양하므로, 제안하는 방법은 제어할 기기와 컨트롤러 모두를 고려하여 사용자 인터페이스를 생성한다. 페이지 안 네트워크를 사용하여 사용자의 상황에 따른 기기의 필요도를 예측하였으며, 행동 선택 네트워크를 사용하여 추상적 사용자 인터페이스를 사용자의 반응에 따라 지속적으로 변할 수 있도록 효과적으로 모델링하였다. 마지막으로 생성된 사용자 인터페이스에 대한 사용자 평가를 수행하여 시스템의 유용성을 확인하였다. ↓

## 참고문헌

- [1] J. Nichols, *et al.*, "Huddle: Automatically generating interfaces for systems of multiple connected appliances," UIST 2006, pp. 279-288, 2006.
- [2] G. D. Kleiter, "Propagating imprecise probabilities in Bayesian networks," *Artificial Intelligence*, vol. 88, no. 1-2, pp. 143-161, 1996.
- [3] B. LaPlant, *et al.*, "The universal remote console: A universal access bus for pervasive computing," *IEEE Pervasive Computing* vol. 3, no. 1, pp. 76-80, 2004.
- [4] J. Nichols and B. A. Myers, "Controlling home and office appliances with smart phones," *IEEE Pervasive Computing*, vol. 5, no. 3, pp. 60-67, 2006.
- [5] J. Nichols, *et al.*, "Demonstrating the viability of automatically generated user interfaces," CHI 2007, pp. 1283-1292, 2007.
- [6] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, pp. 20-24, 2001.
- [7] P. Korpipaa, *et al.*, "Bayesian approach to sensor-based context awareness," *Personal and Ubiquitous Computing*, vol. 7, pp. 113-124, 2003.
- [8] E. Horvitz, *et al.*, *Models of attention in computing and communications: From principles to applications*, Communications of the ACM, vol. 46, no. 3, pp. 52-59, 2003.
- [9] P. Maes, "How to do the right thing," *Connection Science Journal*, vol. 1, no. 3, pp. 291-323, 1989.
- [10] G. F. Cooper and E. A. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309-347, 1992.
- [11] S. Card, T. P. Moran, and A. Newell, *The Psychology of Human Computer Interaction*, 1983.