

시맨틱 네트워크를 이용한 유비쿼터스 가정환경 장치의 입출력 매핑

송인지¹, 홍진혁², 조성배³

연세대학교 컴퓨터과학과

{schunya¹, hjinh²}@sclab.yonsei.ac.kr sbcho@cs.yonsei.ac.kr³

I/O mapping for ubiquitous home devices with semantic networks

In-Jee Song¹, Jin-Hyuk Hong², Sung-Bae Cho³
Dept. of Computer Science, Yonsei University

요약

유비쿼터스 가정환경에서 서비스를 제공하기 위한 다양한 장치들은 각기 고유한 인터페이스를 가진다. 사용자는 이 장치들을 제어하기 위해서 각각 다른 인터페이스에 익숙해야 하며, 결국 장치 수만큼의 인터페이스를 다루어야 한다. 이와 같은 불편을 해소하기 위해서는 하나의 입력 장치로 여러 장치들을 조작하는 사용자 인터페이스가 필요하다. 특히 유비쿼터스 가정환경에서는 다양한 장치들의 상태 및 기능 등이 동적으로 변하고, 장치가 설정되는 환경도 일정하지 않기 때문에 사용자 중심의 유비쿼터스 환경을 제공하기 위해서는 다양한 인터페이스를 통합할 필요가 있다. 사용자가 비슷하게 인지하는 이종 장치들의 기능을 통합하여 사용자 인터페이스의 동일한 입력으로 매핑한다면 사용자의 부담을 줄일 수 있을 것이다. 본 논문에서는 유비쿼터스 가정환경의 다양한 장비들과 인터페이스 사이의 입출력 관계를 분석하여 시맨틱 네트워크로 모델링하는 방법을 제안한다. 각 장치의 상태와 기능을 시맨틱 네트워크로 정의하고, 노드나 엣지 사이의 유사도를 평가하여 장치와 사용자 인터페이스 사이를 자동으로 매핑한다. 제안하는 방법을 가정환경 입출력장치에 적용하고, 입출력 매핑을 시뮬레이션하는 환경을 구현하여 유용성을 검증한다.

Keyword : 유비쿼터스 가정환경, 시맨틱 네트워크, 자동 매핑

1. 서론

가정에서 가전제품을 원격으로 조작하기 위해서 각 제품별로 정의된 리모컨을 사용해야 한다. 가정에 설치되는 전자제품이 늘어날 때마다, 사용자는 다수의 리모컨을 조작해야 한다. 예를 들어, 홈씨어터를 구성하기 위해서는 케이블 셋탑박스, DVD 플레이어, TV, 오디오, 비디오, DivX 플레이어 등의 다양한 장치들을 조작해야 한다. 사용자가 원하는 서비스를 받기 위해서는 적절한 리모컨을 사용해야 하지만, 그림 1 과 같이 리모컨들의 생김새가 유사하기 때문에 사용자는 다수의 리모컨을 선별적으로 사용해야 한다. 뿐만 아니라 버튼의 위치나 모양, 조작하는 방식이 리모컨마다

다르기 때문에 사용자에게 혼란을 초래할 수 있다.

리모컨이 다양한 기능을 수행하기 위해서는 많은 수의 버튼이 필요하다. 제조사는 보통 제품이 가지고 있는 대부분의 기능을 각 버튼에 직접 매핑하기 때문에 기능의 수만큼 버튼이 사용된다. 하지만, 실제로 사용되는 버튼은 약 33% 정도 밖에 되지 않는다[1]. 더욱이 유비쿼터스[2] 환경이 실현되면, TV, 에어컨 등의 기존 가전 제품 외에 조명, 창문과 같은 가정환경의 여러 장치들이 하나의 네트워크로 연결되어, 사용성 문제는 더욱 증폭될 것이다. 유비쿼터스 가정환경에서는 다양한 장치들의 상태나 기능이 자주 변하고, 장치가 설정되는 환경도 개별적이기 때문에 보다 사용자



그림 1. 최신의 홈 씨어터 시스템 조작에 필요한 6가지 리모컨 [1]

중심의 인터페이스가 요구된다.

복잡한 인터페이스를 통합하기 위해서 많은 업체들이 리모컨을 확장하여 해당 장치와 관련된 제품을 조작하도록 하고 있다. 몇몇 TV 리모컨의 경우, 모드 선택을 통하여 TV 뿐 아니라 DVD 플레이어, 비디오 플레이어, 오디오 등의 기기를 조작하는 기능을 제공한다. 하지만, 미리 정의된 기기들에 대해서 사용가능하기 때문에, 새로운 제품이나 잘 사용되지 않는 제품을 조작하기는 매우 어렵다. 최근에는 이런 한계를 극복하기 위해서, 통합 리모컨이나 PDA 용 프로그램 등이 도입되고 있다. 리모컨에서 송출하는 적외선 신호를 감지해 자신이 원하는 버튼에 매핑한다. 이를 위해서는 각 기능과 버튼을 일일이 매핑해야 하고, 통합리모컨이나 PDA 용 인터페이스에 익숙하지 않은 사용자는 사용에 어려움을 겪게 된다.

본 논문에서는 인터페이스와 장치 사이의 관계를 분석하여, 각 인터페이스와 장치를 시맨틱 네트워크로 모델링하는 방법을 제안한다. 시맨틱 네트워크로 표현된 인터페이스와 장치를 각 노드 사이의 유사성과 제약 조건을 사용하여 동적으로 연결한다. 이를 통해 사용자는 유비쿼터스 환경에서 손쉽게 원하는 서비스를 실행시킬 수 있을 것이다.

2. 관련 연구

2-1. 입출력 장치 정의

Universal Serial Bus (USB) Human Interface Device (HID) Usage Table[3] 은 USB 장치를 정의한 USB HID 보고서의 일부분으로 응용프로그램 개발자를 위해 입력장치의 각 컨트롤이 갖는 목적과 데이터 형식을 알려 준다. Usage Table 은 연관되어 사용되는 버튼들을 하나의 묶음으로 처리해, 버튼 집합의 목적과 데이터 형식을 정의할 수 있도록 하였다. 예를 들어, TV 에서 채널을 선택하는 숫자 패드의 경우 0 번 버튼부터 9 번 버튼 까지가 하나의 그룹으로 취급되어 하나의 데이터로 표기된다.

시맨틱 네트워크는 옛지들로 연결된 노드들로 구성된 방향성 그래프이다. 각 노드는 개념들을 표현하고, 옛지는 노드가 의미하는 개념들 사이의 관계를 표시한다. 시맨틱 네트워크는 주로 지식 표상을 위한 형식으로 사용되며, 단순하고, 자연적이고, 명확하며, 가시적인 장점을 갖는다[4]. 시맨틱 네트워크는 계층적인 구조를 갖는 XML 으로 표현할 수 있는데, 이 경우 자료를 조직화하고, 의미를 명확히 하는데 도움이 된다[5].

시맨틱 웹은 의미 정보를 고려하지 않은 기존 웹 문서의 한계점을 극복하고 응용프로그램이 웹

문서들을 의미 수준에서 처리하도록 제안되었다 [6]. 시맨틱 웹에서는 문서가 구조와 데이터의 의미 정보를 표현하는 온톨로지인 시맨틱 네트워크 형태를 갖는다. 응용프로그램은 데이터를 분류하고 데이터 사이의 관계를 표시하며, 데이터가 갖는 속성을 알 수 있어, 원하는 정보를 쉽게 찾고, 통합할 수 있다.

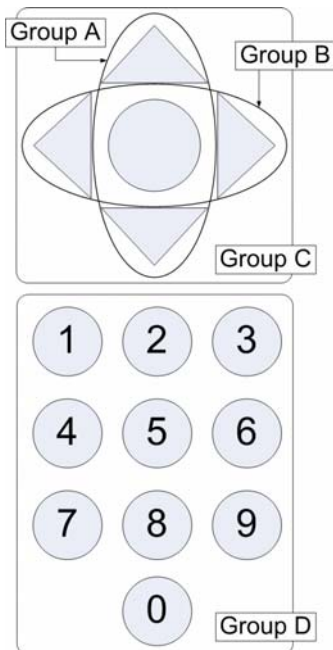
2-2. 입출력 장치의 매핑

Cardoso 는 웹 서비스 검색 과정에서 서비스 객체와 서비스 템플릿을 매칭한다[7]. 서비스 객체(SO)는 시스템이 찾고자 하는 실제 웹 서비스를 나타내고, 서비스 템플릿(ST)은 사용자가 원하는 웹 서비스의 기능을 중심으로 정의된다. SO와 ST는 온톨로지를 사용하여 의미 정보를 표현하기 위하여 시맨틱 네트워크로 설계된다. 매칭은 두 단계로 이루어지는데, 우선 ST와 가장 유사한 SO를 찾는다. 이 과정에서 ST와 SO 사이의 유사도는 구문적 유사도와 기능적 유사도를 사용한다. 구문적 유사도는 ST와 SO의 서비스 이름과 서비

스 설명을 비교하고, 기능적 유사도는 ST와 SO가 갖는 기능 명세를 살핀다. 두 유사도를 바탕으로 ST에 가장 유사한 SO를 찾은 후, ST의 기능과 SO의 기능을 매핑하기 위해 의미적 유사도를 사용한다. 의미적 유사도는 구문적 유사도를 보완하기 위해 사용되는데, ST와 SO를 구성하는 개념과 속성들을 온톨로지를 이용하여 비교한다. 의미적 유사도는 자카드 상관계수(Jaccard coefficient)[8]를 사용하여 구하는데, 자카드 상관계수는 수식 1과 같다.

$$S_{ij} = \frac{p}{p+q+r} \quad (1)$$

단, S_{ij} = 개체 i와 개체 j 사이의 유사도,
 p = i와 j가 모두 가지고 있는 것,
 q = i는 갖고 있고, j는 갖고 있지 않은 것,
 r = i는 갖고 있지 않고, j는 갖고 있는 것.



(a) 인터페이스의 버튼 배열

```

<interface name="example">
  <atom id="0">
    <usage type="nextItem">
      <concept name="menu"/>
      <concept name="up"/>
      <group name="c"/>
    </usage>
    <usage type="nextItem">
      <concept name="channel"/>
      <concept name="up"/>
      <concept name="increase"/>
      <group name="a"/>
    </usage>
  </atom>
  <atom id="1">
    <usage type="previousItem">
      <concept name="menu"/>
      <concept name="left"/>
      <group name="c"/>
    </usage>
    <usage type="previousItem">
      <concept name="volume"/>
      <concept name="down"/>
      <concept name="decrease"/>
      <group name="b"/>
    </usage>
  </atom>
  <atom id="2">
    <usage type="select">
      <concept name="confirm"/>
      <concept name="menu"/>
      <group name="c"/>
    </usage>
  </atom>
  ... 중략 ...
  <atom id="5">
    <usage type="combination">
      <concept name="number"/>
      <concept name="1"/>
      <group name="d"/>
    </usage>
  </atom>
  ... 중략 ...
</interface>

```

(b) (a)의 버튼 배열을 표현하는 XML

그림 2. 입력 인터페이스 예제

3. 가정환경 장치의 입출력 매핑

3-1. 입출력 장치 정의

입력장치의 온톨로지를 나타내는 시맨틱 네트워크는 XML 을 이용하여 표현하였다. 각 입력 인터페이스는 하나의 인터페이스 노드를 갖고, 인터페이스 노드는 버튼을 나타내는 atom 노드를 갖는다. 그림 2 와 같은 인터페이스의 경우 하나의 버튼이 메뉴 내비게이션에 사용되기도 하고, 채널 조작에 사용 되기도 한다. 하나의 버튼이 여러 가지 용도로 사용되는 경우에는 그림 2(b)의 id 가 0 인 atom 노드와 같이 다수의 usage 노드를 갖는다. 사용 처를 표현하는 usage 노드는 USB HID usage table 에서 사용된 세 가지 기본요소 버튼의 사용 목적에 대한 개념, 타입과 함께 사용되는 버튼 그룹을 각각 concept, type, group 노드를 속성으로 정의한다. 여기서 사용 목적의 타입은 버튼의 사용 목적을 분석하여 표 1 와 같이 네 가지로 분류한다. 예를 들어, TV 채널을 선택하는 버튼 중에서 현재 채널의 다음 채널을 선택하는 버튼은 nextItem, 이전 채널을 선택하는 버튼은 previousItem, 10 번 채널을 선택하는 버튼의 경우 select 를 타입으로 갖는다. 그림 2(a)의 숫자 패드와 같은 경우는 그룹 내의 다른 버튼과 함께 사용되어 채널을 선택하므로 그림 2(a)의 id 가 5 인 atom 노드에서 보듯이 combination 타입을 갖는다.

출력장치는 장치가 갖는 속성들과 속성에 포함되는 상태의 집합으로 표현된다. Brightness 와 power 의 두 가지 속성을 갖는 조명 장치는 표 2 와 같이 표현될 수 있다. 속성은 상태를 두 가지 방식으로 정의하는데, brightness 속성의 경우와 같이 상태가 자연수 집합의 일부분인 경우는 statusRange 노드를 사용하여 표현한다. Power 속성

표 1. 버튼 사용의 종류

종류	설명
nextItem	다음 상태를 선택
previousItem	이전 상태를 선택
select	특정 상태를 선택
combination	다른 버튼과 함께 사용되어 특정 상태를 선택

표 2. 조명 장치를 표현하는 XML

```

<functions name="light">
  <property name="brightness">
    <statusRange from="1" to="100"/>
  </property>
  <property name="power">
    <status name="on"/>
    <status name="off"/>
  </property>
</functions>

```

과 같이 상태가 수로 표현되지 않고 개념들의 집합으로 이루어지는 경우 status 노드를 이용해서 각 상태를 표현한다.

3-2. 입출력 장치 매핑

그림 3 은 입력장치와 출력장치의 매핑과정을 보여준다. 출력장치의 속성을 상태 수가 많은 순으로 정렬하고, 상태 수가 많은 출력장치의 속성부터 입력장치의 버튼과 매핑한다. 매핑 과정에서 출력장치의 속성의 상태 수와 버튼 수, 버튼 그룹의 수를 고려하여 속성을 하나의 버튼과 매핑하게

FuncProperties ← 매핑되지 않은 속성 목록

UIAtoms ← 매핑되지 않은 버튼 목록

in ← UIAtoms.Count

out ← FuncProperties.Count

WHILE (in > 0 and out > 0)

IF (in ≤ out)

매핑 1: 버튼 하나와 속성 하나 매핑

ELSE

outState ← 매핑되지 않은 모든 상태의 수

inGroup ← 매핑되지 않은 버튼 그룹의 수

IF (in ≤ outState and inGroup > 0)

매핑 2: 버튼 그룹과 속성 하나를 매핑

ELSE

매핑 3: 여러 버튼과 속성 하나를 매핑

in ← UIAtoms.Count

out ← FuncProperties.Count

그림 3. 입·출력장치 매핑 알고리즘

나, 하나의 버튼그룹과 매핑한다. 사용할 수 있는 버튼의 수가 많은 경우 속성의 상태마다 버튼을 할당한다. 속성과 버튼 하나를 매핑하는 경우에는 의미적 유사성을 자카드 상관계수를 사용하여 결정하고, 유사성이 가장 큰 버튼을 선택하여 속성과 `nextItem` 타입으로 매핑한다. 속성과 버튼 그룹 하나를 매핑하는 경우에는 우선적으로 매핑하는 속성은 상태의 수가 많은 속성이므로 버튼 그룹 중에서 버튼 그룹을 구성하는 버튼 수가 가장 많은 그룹을 선택하여 해당 차례의 속성과 매핑한다. 이 과정에서 버튼의 타입은 선택된 버튼 그룹에서 버튼의 사용 타입에 따라 결정된다. 여분의 버튼이 많아 속성의 각 상태마다 버튼을 매핑하는 경우에는 첫 번째 매핑 방법과 유사하게 상태와 버튼의 자카드 상관계수를 이용하여, 유사성이 가장 높은 상태와 버튼을 `select` 타입으로 매핑한다.

4. 실험 결과

4-1. 실험 환경

입력장치와 출력장치의 매핑 실험을 위해서, Microsoft .Net 환경 하에서 C#으로 매핑 인터페이스를 구현하였다. 입력장치와 출력장치의 의미 정

보를 그림 2 (b)나 표 2 와 같은 XML 형식으로 표현하여 프로그램에서 불러오면, 오른쪽 테이블에 입력장치의 버튼과 출력장치의 기능의 매핑 결과를 출력한다. 입력장치와 출력장치 사이의 매핑 결과를 평가하기 위해 TV 리모컨과 TV, 오디오를 시맨틱 네트워크로 설계하였다. TV 리모컨은 그림 4 와 같이, TV 와 오디오는 그림 5 과 같이 설계하였다.

4-2. 입출력 장치 매핑 결과

TV 리모컨과 TV 및 오디오를 매핑한 결과는 그림 5 과 같다. TV 리모컨과 TV 사이의 매핑은 각 노드 사이의 의미적 유사도가 큰 것들이 많아서, 대부분 사용성에 큰 영향이 없는 매핑이 이루어졌다. TV 리모컨과 오디오 사이의 매핑도 전원과 채널의 경우 의미적 유사도를 갖으므로 사용성에 문제가 적은 매핑이 이루어졌지만, 의미적 유사도가 전혀 없는 경우 속성의 수와 버튼의 수만을 고려하기 때문에 음량 조절 기능이 버튼 그룹 D 에 매핑되고, 재생 기능이 버튼 그룹 B 에 매핑되어 사용성이 낮은 매핑 결과가 발생하였다.

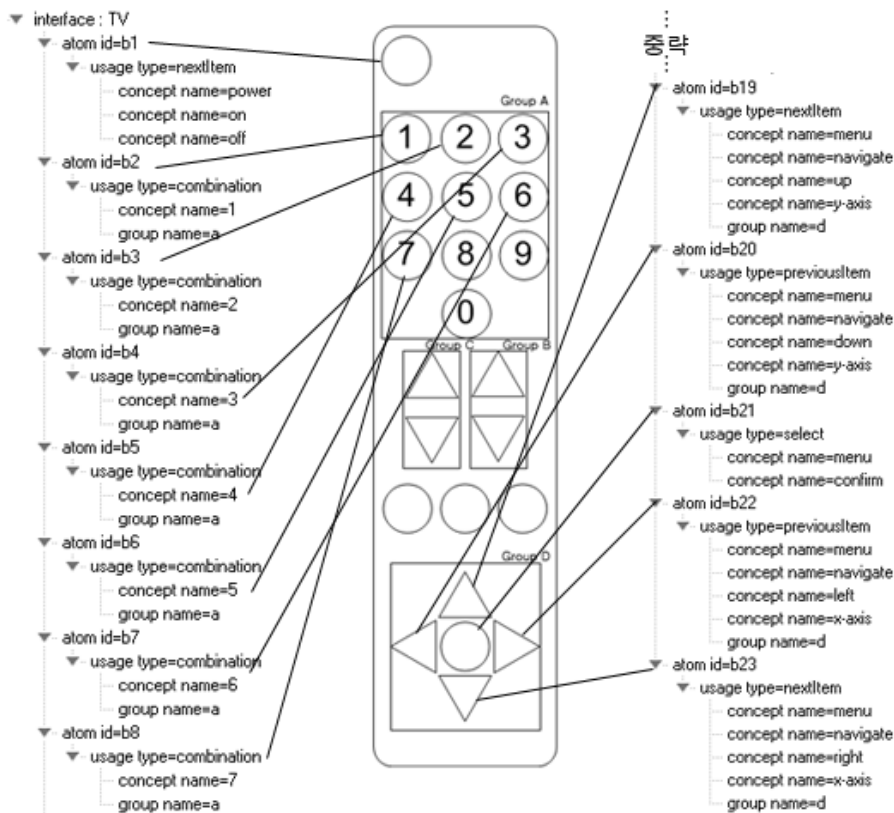


그림 4. TV 리모컨의 시맨틱 네트워크 표현

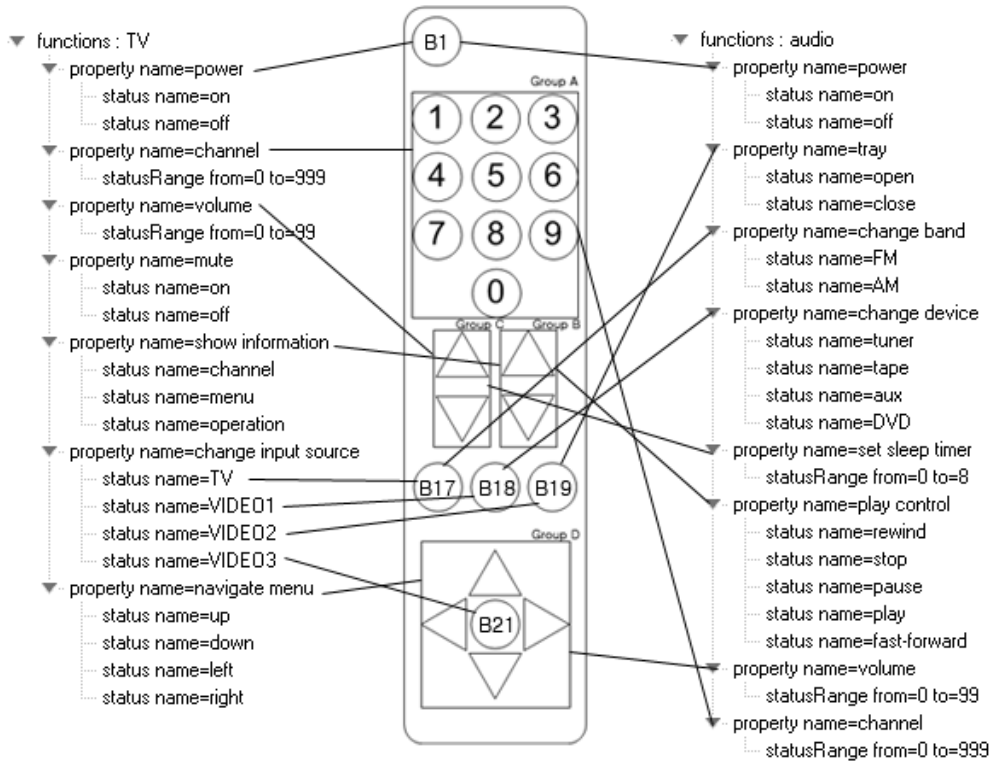


그림 5. TV 리모컨과 TV 및 오디오와의 매핑 결과

5. 결론 및 향후과제

본 논문에서는 사용자가 유비쿼터스 환경에서 통합된 사용자 인터페이스를 통해 원하는 서비스를 실행시키도록, 각 인터페이스와 장치를 시맨틱 네트워크로 모델링하고 각 노드 사이를 의미적 유사도와 제약 조건을 사용하여 매핑하는 방법을 제안하였다. 의미적 유사도를 구하기 어려운 경우 매핑의 유효성이 감소하므로, 향후에는 사용자 모델링의 추가를 통해 사용자 선호도를 반영하여 의미적 유사도를 보완할 필요가 있다. 또한, 사용성 평가를 통하여 제안하는 방법의 유효성을 정량적으로 검증할 필요가 있다.

6. 감사의 글

본 연구는 지능형 HCI 융합연구센터의 지원을 받아 진행 되었습니다.

7. 참고 문헌

[1] J. Nielsen, "Remote control anarchy," <http://www.useit.com/alertbox/20040607.html>

[2] M. Weiser, "The Computer for the 21st Century," Scientific American, vol. 265, no. 9, pp. 66-75,

September 1991.

[3] B. Bates et al., "USB HID Usage Table," Revision 1.12, Jan 2005. [Online], Available: http://www.usb.org/developers/devclass_docs/Hut1_12.pdf

[4] M. Marinov and I. Zheliazkova, "An interactive tool based on priority semantic networks," Knowledge-Based Systems, no. 18, pp. 71-77, 2005

[5] L. Feng, E. Chang and T. Dillon, "A semantic network-based design methodology for XML documents," ACM Trans. on Information System, vol. 20, no. 4, pp. 390-421, October 2002

[6] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," Scientific American, no. 279, pp. 34-43, May 2001

[7] J. Cardoso and A. Sheth "Semantic E-Workflow Composition," Journal of Intelligent Information Systems, vol. 21, no. 3, pp. 191-225, 2003.

[8] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, CA, 2001.