

유비쿼터스 컴퓨팅 환경에서 동작하는 서비스의 사용자 만족도 평가 방법¹⁾

이건수[○] 김민구

아주대학교 정보통신 전문 대학원
lks7256@ajou.ac.kr, minkoo@ajou.ac.kr

Evaluating Method of User's Satisfaction for the Services in Ubiquitous Computing Environment

Keonsoo Lee[○] Minkoo Kim

Graduate School of Information and Communication, Ajou University

요 약

유비쿼터스 컴퓨팅 환경에서 서비스는 환경 정보를 감지하고 그 정보를 바탕으로 대상 사용자에게 서비스를 제공해준다. 이때, 수행되는 서비스는 사용자로부터 명시적인 요청을 받아 수행되는 것이 아니라, 주어진 상황을 기반으로 자동으로 수행되는 것이기 때문에, 이 서비스가 사용자가 정말 원하는 서비스인가를 판단하는 것은 서비스를 수행하는 시스템에 대한 신뢰도 평가에 중요한 기준이 된다. 이에 본 논문에서는 사용자의 의도에 맞는 서비스 수행을 보장하기 위해, 서비스에 대한 사용자의 만족도를 평가하는 방법을 제안한다. 제안된 방법은 OCC 감정 모델을 기반으로 사용자의 서비스 만족도를 계산하고, 이 결과를 통해 사용자의 의도와 서비스의 수행 조건을 일치시킨다. 만족도 계산 작업의 자동화를 통해 서비스의 수행 조건은 사용자의 의도 변화에 따라 동적으로 변경 가능하고, 이를 통해 사용자는 시스템에 대한 하위수준의 조정 작업 없이 자신의 행동 변화만으로 서비스의 실행 조건을 변경시킬 수 있다.

1. 서 론

상황 인지는 유비쿼터스 컴퓨팅 환경에서 지능화된 서비스를 제공하기 위한 기본 요소이다. 상황 인지를 통해서, 서비스는 스스로 언제 동작해야 하는지 또 어떻게 동작해야 하는지를 판단할 수 있게 된다[1]. 가령, 상황 인지 기능이 있는 핸드폰이라면 사용자의 현재 위치 및 행동을 상황 정보로 인식하여 사용자가 회의실에서 앉아있다는 정보를 인지해 이 정보로부터 사용자가 현재 회의 중이라는 상황을 판단해 전화벨을 진동으로 변경하거나, 수신 전화를 음성 사서함으로 옮기는 등의 그 상황에 적합한 서비스를 수행할 수 있게 된다. 이처럼 상황 인지 기능은 사용자로 하여금 명시적으로 하나하나의 상세 정보를 기술하지 않더라도 일정 수준의 지능을 갖고 환경에 능동적으로 대처하는 서비스를 구현할 수 있도록 해 준다. 서비스가 스스로의 판단으로 수행되는 과정에서는 서비스 자체에 대한 정확도는 중요한 평가 요소가 된다. 서비스에 대한 정확도는 2가지 측면에서 바라볼 수 있는데, 첫 번째 측면은, 그 서비스가 자신의 목표를 달성하기 위해 동작하는 과정에서의 정확도이다. 온도를 낮춰주는 서비스의 경우, 목표 온도로까지 온도가 낮춰져야 하고, 길안내 서비스의 경우, 목표 장소까지 안내가 지속되어야 한다. 두 번째 측면은, 과연 그 서비스가 주어진 환경에서 동작해야 하는 서비스를 선택하는

과정에서의 정확도이다. 가령 라이터를 켜더니 화재경보기가 작동한다던지, 피곤해서 소파에 앉았을 뿐인데, TV가 켜진다던지 하는 경우는 두 번째 측면에서의 정확도가 보장되지 못한 경우이다.

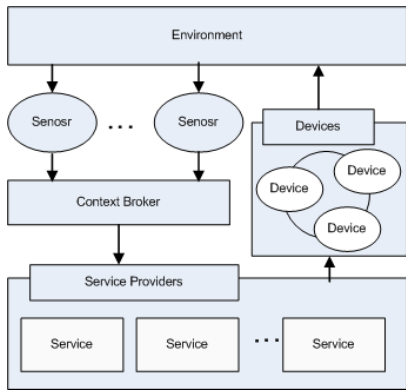
따라서, 유비쿼터스 컴퓨팅 환경에서 제공되는 서비스의 정확도를 확인하기 위한 방법이 필요하다. 서비스의 정확도는 서비스가 초기의 목표를 올바르게 달성했는가와 그 서비스가 정확한 상황에서 발현되었는가를 기준으로 평가된다. 첫 번째 기준은 서비스의 수행 과정을 검토함으로써 측정되는데, 목표를 달성해 나가는 과정에서 사용되는 장치들 사이에서의 충돌 해결 개별 장치의 사용에 있어 목표와 상반되는 동작의 발생 목표 도달까지의 소요 시간 등을 고려하게 된다. 반면, 두 번째 기준은, 기본적으로 특정 서비스에 대한 사용자의 선호도 프로파일 정보를 바탕으로 평가하는데 이때 사용되는 사용자의 선호도 프로파일은 서비스의 목표를 설정하기 위한 정보와, 서비스가 자체적으로 인지한 상황이 그 사용자가 서비스를 필요로 하는 상황과 일치하는지를 판단하기 위한 정보로 구성된다. 이상의 두 가지 평가 결과는 주어진 환경에서 사용자에게 정확한 서비스를 제공하기 위한 토대로 사용될 수 있다.

2. 관련 연구

상황 인지 시스템은 일반적으로 <그림 1>의 구조를 갖고 있다[2]. 환경 정보는 센서에 의해서 수집되고, 컨텍스트 브로커는 수집된 정보를 바탕으로 의미 정보를 추출해내, 서비스 제공자에게 전달해 주면 각각의 서비

1) 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업의 08B3-S2-10M 과제로 지원된 것임

스 제공자는 현 환경안에서 자신이 수행되어야 하는지를 판단하고, 공간에 존재하는 장치들을 사용해 서비스를 수행한다. 이러한 서비스의 수행은 다시 환경에 변화를 야기시키고, 이러한 변화는 다시 센서들에 의해 인지되어 또 다른 서비스의 발생을 촉진시키게 된다 물론, 각 서비스는 컨텍스트 브로커를 거치지 않고 직접 센싱 정보를 관리할 수 있고, 각각의 장치들은 개별 서비스에 종속될 수도 있지만, <그림 1>의 각 세부 요소들이 수행하는 기능은 반드시 포함하고 있어야 한다



<그림 1> 일반적인 상황 인지 시스템의 구조

이렇게 구성된 상황인지 시스템은 스스로의 동작 조건에 대한 지식을 관리하기 위해 규칙 기반 시스템에 기반한 선결조건(pre-condition)/후결 조건(post-condition)을 사용한다[3]. 이러한 방법은 서비스의 발생 조건으로 사용될 뿐만 아니라, 서비스를 수행하는 과정에서 필요한 장치들을 선택하고, 선택된 장치들의 사용 시 발생하는 연쇄 조건을 기술하기 위해 사용된다[4]. 가령, 모닝콜 서비스를 제공하는 시스템이 존재한다고 가정해 보자. 서비스는 아침이 되어도 사용자가 아직 자고 있다면 이 서비스는 그 사용자를 깨워준다 잠을 깨우기 위해, 서비스는 알람을 울리고 조명을 밝히고, 창을 커튼을 열고, 환기를 시켜주는 일련의 작업을 수행한다고 하자 이러한 서비스의 수행 규칙은 <표 1>처럼 만들 수 있다. 이 표에 근거하여 서비스는 특정 환경에서 동작하고 환경을 목표한 상황으로 변경시킴으로써 종료된다 또한 이 과정에서 사용되는 개별 장치들의 동작 흐름을 결정할 수 있다.

<표 1> 서비스의 선/후결 조건의 예

Service	Pre-Condition	Post-Condition
모닝콜 서비스	아침 \cap 수면중	사용자의 기상
Device		
알람	아침 \cap 수면중	사용자의 기상
조명	알람 \cup (아침 \cap 조도 < 생활조도)	사용자의 기상 \cap (조도 > 생활조도)
환기장치	알람 \cup (impurity of air > value)	impurity of air < value
커튼	(intensity of illumination of indoor < that of outdoor) \cup 아침	NULL

이때, 서비스는 정해진 선결 조건에 의해 발현되는데 이 선결조건이 사용자의 의도에 부합되는가에 대한 문제가 발생한다. 즉, 알람 서비스가 동작한다고 가정했을 때, 이 서비스는 사전에 정의된 선결 조건 즉 아침이라는 시간과 현재 사용자가 자고 있다는 정보를 자신의 상황 정보로 인지해 수행을 결정한다 그러나, 만약 휴일이어서 일찍 일어날 필요가 없거나, 다른 일정으로 인해 보다 일찍 일어나야 하는 경우, 이 서비스의 수행은 사용자가 원하는 서비스를 제공하지 못하게 되고 이 같은 서비스 수행의 오류는 사용자로 하여금 그 시스템에 대한 강한 불신을 야기시키는 결과를 가져온다 이는 곧 서비스의 수행 과정에 있어서 올바른 상황 인지도 중요하지만 이와 함께, 서비스가 수행 되어야 하는지를 올바르게 결정할 수 있는 것 역시 중요함을 의미하고 있다

3. 제안 방법

본 논문에서는 유비쿼터스 컴퓨팅 환경에서 사용자에게 제공되는 서비스에 대한 사용자 만족도를 평가하기 위한 방법을 제안한다 사용자의 만족도는 서비스의 올바른 수행 결과와 함께, 그 서비스가 사용자의 요구에 부합되는지에 대한 판단으로 결정된다 서비스의 올바른 수행 결과에 대한 부분은 서비스를 정의하고 그 동작 프로세스를 기술하는 과정에서 명시된 내용과 부합되면 서비스가 올바르게 수행되었다고 볼 수 있기 때문에 기존의 소프트웨어 테스트 기법을 동일하게 적용하여 확인할 수 있다. 즉, 서비스의 수행 결과가 서비스의 목표 상황과 일치한다면, 그 서비스는 성공적으로 수행되었고 수행 과정에서 서비스 스펙에 기술된 내용과 동일하게 동작했다면, 그 서비스는 정확한 동작을 보였다고 평가할 수 있다[5]. 반면, 그 서비스가 주어진 상황에서 사용자의 요구에 부합하는 서비스였다는 것을 평가하는 것보다 추상적인 평가 기준을 갖는데 이는 사용자의 의도나 요구를 명시적으로 표현하기가 어렵고 각 서비스의 수행 목표와 사용자의 특정 의도를 맞추는 것 역시 모호한 작업이기 때문이다 더욱이 유비쿼터스 컴퓨팅 환경에서의 상황 기반 서비스라는 특성은 기존의 소프트웨어 만족도 평가 방법을 차용함에 있어 제약으로 작용하는데, 이는 기존의 만족도 평가를 위한 과정이 시간과 비용을 많이 소모하기 때문이고 이는 유비쿼터스 시스템에 대한 사용자의 기대에 반하는 일이기 때문이다 즉 사용자들을 대상으로 설문지를 만들어 배포하고 작성된 설문지를 수집하여 해석하고 그 결과를 반영하여 시스템을 변경한 뒤 재배포의 과정이 이루어지기도 전에 이미 사용자는 서비스에 대한 불만을 토로하면서 시스템을 불신하게 된다. 이러한 이유로 대부분의 상황인지 서비스들이 추천 시스템의 형태를 따라 사용자에게 서비스 수행에 대한 책임을 전가하고 있다[6]. 이 경우, 사용자는 자신의 상황에서 제공 받을 수 있는 서비스를 시스템으로부터 추천받고, 그 서비스들 중에서 자신의 명시적인 선택으로 서비스의 수행을 결정하게 되고 이는 사용자의 의도/요구에 부합하지 않는 서비스의 구동을 원천적으로 방지할 수 있다. 그러나, 끊임없이 발생하는 추천의 빈도는 시스템의 복잡해질수록 사용자의 만족도를 떨어

뜨리게 된다. 가령, 특정 상황을 선결조건으로 갖고 있는 서비스가 N개 존재한다고 하면 그 상황에서 사용자는 N개의 서비스에 대한 추천을 받게 되고 N번 서비스를 받을 것인지 아닌지를 선택해야한다 더욱이 상황 변화가 빈번하게 일어날수록, 각각의 상황에 대한 서비스 추천이 매번 일어나게 되고 사용자는 끊임없이 각 서비스에 대한 추천에 대한 명시적 응답을 표현해야 한다

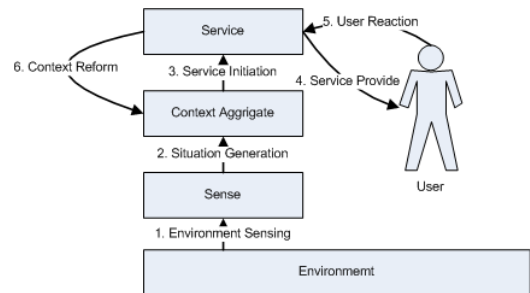
이에 본 논문에서 제안하는 방법은 서비스 사용자의 감정과 행동을 근거로 서비스의 선행 조건에 대한 평가를 수행한다. 사용자의 감정을 평가하기 위해 OCC 감정 모델을 기반으로 사용한다[7]. OCC 모델은 사건, 행위자, 대상에 대한 감정의 종류와 그들 사이의 관계를 모델화 시킨 구조를 갖고 있는데 <표 2>는 OCC 모델의 구조를 보여준다.

<표 2> OCC 감정 모델

대상	감정 집합	감정
사건	Fortunes of others	happy-for gloating resentment pity
	Prospect-based	satisfaction fear-confirmed relief disappointment
	Well-being	joy distress
행위자	Attribution	pride shame admiration reproach
	Well-being/Attribution Compound	gratification gratitude remorse anger
대상	Attraction	love hate

이 중, 서비스에 대한 사용자의 만족도를 평가하기 위해 사용하는 감정 집합은 사건에 대한 자신의 만족도를 표현하는 Prospect-based 감정 집합을 사용한다 이 감정 집합이 의미하는 내용은 다음과 같다 어떤 사건이 발생했을 때, 그 사건에 대해 느끼는 감정은 욕망(desirability), 기대(likelihood), 그리고 사건의 결과(effort realization)로 구성된다는 것이다 가령, 어떤 사람이 한적한 밤길을 걸어간다고 가정해 보자 이 사람은 목적지까지 안전하게 도착하고 싶다는 욕망을 갖고 있다. 이때, 뒤에서 발자국 소리가 들려온다 이 사람은 발자국의 주인공이 강도인지 그냥 지나가는 사람인지에 대한 기대를 한다. 얼마 전 같은 장소에서 강도가 일어났었다는 것을 알고 있다면 강도가 쫓아오고 있을 것이라는 예상을 할 것이고 그렇지 않다면, 그냥 지나가는 사람일 것이라는 기대를 하게 될 것이다 그리고, 마침내, 그 발자국 소리의 주인공이 자기 옆을 스쳐 지나갔다고 할 때, 이 사람이 강도를 예상했다면 안심(relief)

할 것이고, 그냥 지나가는 행인을 예상했다면 자신의 예상에 대한 만족(satisfaction)을 할 것이다. 반면, 정말 강도였을 때, 이 사람이 강도를 예상했다면 불안감을 확인(fear-confirmed)했을 것이고, 행인을 예상했다면 실망(disappointment)하게 되었을 것이다 이러한 감정 변화 모델을 차용했을 때 서비스에 대한 사용자의 감정 변화는 다음과 같이 정의할 수 있다 특정 상황에서 사용자는 그 상황의 변화를 욕망하게 되고 원하는 상황으로의 변화를 가져올 수 있는 서비스를 기대하게 된다 이 때, 사용자가 처한 상황을 인지한 시스템은 자신의 인지 내용에 따라 그 상황에 적합한 서비스를 자동으로 실행시킨다. 서비스의 수행에 의해 상황은 변화하게 되고, 변화된 상황이 사용자가 의도한 상황과 일치하면 만족 혹은 확인의 감정을 불러일으키고 그렇지 않다면, 안심 혹은 실망의 감정을 생성한다 이때, 사용자의 서비스에 대한 기대는 항상 긍정적이라고 가정하면 서비스의 수행으로 발생하는 사용자의 감정은 만족 혹은 실망으로 축소시킬 수 있다. 이는 사용자가 원치 않는 서비스가 수행될 것이라는 기대를 하지 않는다고 가정할 수 있기 때문이다. 가령, 사용자가 잠자리에 들면서 “혹시 텔레비전이 자동으로 켜지는 것 아니야?”라는 식의 기대는 고려대상에서 제외할 수 있기 때문이고 이것이 가능한 이유는 텔레비전을 작동시키는 서비스가 인지하는 선결 조건으로 “사용자가 잔다”라는 요소를 넣지 않기 때문이다 다시 말해, 본 논문에서 제안하는 방법은 주어진 상황에서 동작하는 서비스가 사용자의 기대에 부응하는 서비스냐를 판단하는 방법으로 상황이 주어졌을 때, 그 상황에 맞지 않는 서비스는 수행되지 않음을 기본 가정으로 하고 있다.



<그림 2> 만족도에 따른 서비스 변화과정

서비스의 수행 결과로 야기된 사용자의 감정이 만족이라면, 그 서비스는 올바르게 수행되었다고 볼 수 있고 사용자의 감정이 실망이라면 그 서비스는 잘못 수행된 것으로 평가된다. 이때, 사용자의 감정 상황을 판단할 수 있는 기준으로서, 서비스에 대한 사용자의 반응 행동이 사용된다. 가령, 사용자가 방에 들어왔다는 상황으로 조명이 켜졌는데, 사용자가 즉시 조명을 껐다면 이는 조명 서비스가 사용자에게 실망의 감정을 야기 시켰다고 볼 수 있다. 이처럼, 서비스가 사용자의 의도를 반영하지 못했다면, 서비스는 옳지 않은 상황에서 발현된 것이고 이는 그 서비스가 갖고 있는 선결 조건이 잘못되었다고 판단할 수 있다. 따라서 서비스의 선결 조건이 변화되어야 하고, 새로운 선결조건은 <그림 1>의 Context Broker

가 관리하고 있는 다른 상황 요소를 추가함으로써 확장되게 된다. <그림 2>는 사용자의 만족도에 따라 서비스 선결 조건의 변화 과정을 보여준다 이때, 서비스의 선결 조건을 확장시키기 위한 방법은 다음과 같다 사용자의 만족도가 실망이 되었을 때, 즉 서비스 수행에 참여한 장치들의 동작을 사용자가 명시적으로 중지시켰을 경우 그 서비스의 상황을 인지한 순간의 다른 센싱 정보를 검사하고, 반복되는 서비스 만족도의 실망 경우에서 동일하게 발생하는 센싱 정보를 그 서비스의 선결 조건으로 포함시킨다. 이에 대한 알고리즘은 <표 3>에 나타나 있고, 이 과정을 통해, 사용자의 만족도 결과는 서비스의 선결 조건을 사용자의 의도에 맞도록 변화시키게 된다

<표 3> 서비스 선결조건 변경 알고리즘

```

PreConditionChange (fail)
//fail : 선결조건 변경까지의 실망횟수
{
    While(fail != itr) {
        user_satisfaction = service_perform();
        //사용자의 서비스 만족도 결과를 받아옴
        if (user_satisfaction == true)
            context_set = environment_current_context;
        //원래 서비스 선결 조건 발생 시의 환경 상황
        else if (user_satisfaction == false)
            { itr++;
              new_context_set = environment_current_context;
            //서비스 실패시의 환경 상황을 받아옴
              for(i=0;i<context_set.length;i++) {
                  if(context_set[i] == new_context_set[i])
                      context_set[i].index++
                }
            //서비스가 실패했을 하거나 성공했을 때, 공통점을 갖고
            //있는 환경 상황 요소를 찾아냄
            }
            if (context_set[dataID].index == itr)
                ServicePreConditon.add(context_set[dataID])
            //공통점을 갖고 있는 환경 상황 요소를 그 서비스의 선결
            //조건으로 추가
        }
    }
}
    
```

4. 실험

본 논문에서 제안한 사용자 만족도 평가 방법을 <표 1>에 적용하였고, 추가 환경 정보로 날짜를 삽입하였다 알람 서비스는 매일 아침 7시에 동작하고, 실망횟수는 8 회로 가정하였다. 사용자는 매 일요일마다 알람서비스에 실망하였다. 이때 실망했음을 확인하기 위한 행동으로 사용자는 방의 조도를 다시 낮췄는데 이를 위해, 전등을 소등하기도 하거나 커튼을 다시 치는 행동을 하였다. 각 행동은 동시에 일어날 수도 있고, 개별적으로 일어날 수도 있는데, 본 실험에서는 장치의 직접적인 상태 변화를

검사하지 않고, 각 장치의 후결 조건의 변화를 비교하였다. 이는 각 장치들이 공유하는 후결 조건이 존재하기 때문에 비교과정을 줄일 수 있었다. 2달 후, 즉 실망 횟수가 지정한 값을 넘어섰을 때, 알람서비스의 선결 조건은 <아침 7시 ~ 수면중 ~ (일요일)>로 변경되었다.

5. 결론

유비쿼터스 컴퓨팅 환경은 주어진 현재 상황에서 필요한 서비스를 자동으로 인지해 사용자에게 제공해 줄 수 있어야 한다. 이를 위해서 상황 인지 기법에 관한 연구들이 수행되어 왔지만, 현재 상황에서 수행 가능한 서비스와 그 상황에서 사용자가 원하는 서비스를 일치시키기 위한 방법으로는 충분하지 않다. 이를 위해 본 논문에서는 사용자의 서비스 만족도를 그 사용자의 감정과 행동을 기준으로 평가하고 그 결과에 따라 서비스의 선결 조건을 사용자의 의도에 부합할 수 있도록 확장시키는 방법을 제안하였다. 사용자의 만족도를 보장하지 못하는 서비스의 선결 조건은 그 사용자의 의도를 반영할 수 있는 상황의 상세화를 통해 변경되는데 이 과정을 자동화시킴으로써, 사용자는 서비스를 하위 수준에서 변경하지 않고서도 서비스의 수행 조건을 변화시킬 수 있다. 제안된 방법은, 관리자에 의한 시스템 변경 없이 사용자의 서비스 사용 행동에 따른 동적인 서비스 조절이 가능하게 해 줌으로써, 유비쿼터스 컴퓨팅 환경에서 사용자들의 편의를 보다 높여줄 수 있다.

Reference

- [1] B. Schmidt, A. Aidoo, K.A. Takaluoma, A. Tuomela, U. Van Laerhoven, K. Van de Velde W. "Advanced Interaction in Context". 1th International Symposium on Handheld and Ubiquitous Computing (HUC99), Springer LNCS, Vol. 1707: 89-101.
- [2] C. Harry, "An Intelligent Broker Architecture for Pervasive Context-Aware Systems" phdThesis, University of Maryland, Baltimore County, 2004.12
- [3] Meyer, Bertrand "Applying "Design by Contract", in Computer" (IEEE), 25, 10, October 1992, pages 40-51
- [4] H. Levesque, F. Pirri, and R. Reiter. "Foundations for the situation calculus" Electronic Transactions on Artificial Intelligence, 2(3-4):159-178. 1998
- [5] Laycock, G. T. "The Theory and Practice of Specification Based Software Testing". Dept of Computer Science, Sheffield University, UK.
- [6] B. Schilit, N. Adams, and R. Want. "Context-aware computing applications". IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), Santa Cruz, CA, US: 89-101.
- [7] Andrew Ortony, Gerald L. Clore, and Allan Collins "The Cognitive Structure of Emotions" Cambridge university press, ISBN 521-38664-0