

U-헬스 시스템을 위한 상황정보 처리 기법 개발

김정석*, 박성도*, 정회영**, 최의인*

*한남대학교 컴퓨터공학과

*{jskim, sdpark, eichoi}@dmlab.hannam.ac.kr

**jhy@ubnc.net

Context Processing Scheme for U-healthcare system

Jeong-Seok Kim*, Sung-Do Park*, Jung Hoi-Young**, Eui-In Choi*

*Dept of Computer Engineering, Hannam University

**UBNC Co., Ltd

요 약

u-헬스케어 서비스는 언제, 어디서, 어떠한 상황에서도 사용자에게 적절한 서비스를 제공할 수 있어야 한다. 그러한 u-헬스케어 서비스를 제공하기 위해서는 센서로부터 수집되는 데이터들에서 사용자의 상황에 적절한 정보인지, 그리고 데이터들의 유효성과 중복성을 검사해야 한다. 또한 이렇게 수집된 정보들을 u-헬스케어 서비스를 제공하는 시스템에서 적절하고 효과적인 관리 및 분석이 되어야만 사용자에게 적절한 헬스케어 서비스를 제공할 수 있게 된다. 따라서 본 논문에서는 사용자에게 적절한 의료 서비스를 제공하는 u-헬스케어 시스템을 위하여 각종 상황정보를 검사하여 데이터를 추출하고, 이러한 데이터를 정형화하기 위한 모델링 기법을 제안한다.

1. 서론

유비쿼터스 컴퓨팅은 현실에 존재하는 모든 대상물들을 연결해 필요한 정보와 서비스를 언제 어디서나 사용자가 원할 때마다 즉시 제공할 수 있는 기반 기술로 정의된다. 유비쿼터스 컴퓨팅은 기술 중심의 발전에서 사용자 중심의 발전으로 패러다임의 변화를 가져왔다. 특히 유비쿼터스 컴퓨팅 환경이 사용자 중심의 서비스를 제공하기 위해서 사용자 및 사용자 주변 환경에 대한 정보를 분석하여 사용자의 필요성을 파악하는 상황인식(context-aware) 기술이 가장 핵심적인 기술 요소로 자리 잡고 있다. 또한 상황인식 기술을 바탕으로 사용자의 성향 및 주변 환경 등에 대한 정보를 수집, 분석하여 개개인에게 가장 적합한 서비스를 제공하는 것이 유비쿼터스 컴퓨팅의 가장 핵심적인 목표이다.

그 중 u-헬스케어 시스템은 사회적 측면에서 급속한 고령화 현상, 웰빙 열풍과 시장 환경 측면에서는 보건비 지출증가, 만성질환자 증가, 바이오 산업의 활성화 등에 적지 않은 과장이 미칠 것으로 예상되고 있다. 사용자의 적극적인 참여가 예상되면서 u-헬스 산업이 사용자 중심으로 변화하는데 주도적인 역할을 하게 될 것이다. 또한 IT 인프라의 급속한 발전이 u-헬스 측면의 미래를 밝게 예측하도록 선두자적인 역할을 하게 될 것으로 예상된다.

이러한 u-헬스케어에서의 상황인식 컴퓨팅이 이루어지기 위해서는 상황과 관련된 많은 데이터들이 있다고 했을 때 이들 중에서 실제로 상황을 분류하고 규정하는데 쓰일 수 있는 데이터를 찾아 정제하는 과정이 필요하다. 그리고

정제 과정을 통해 상황 데이터로 분류된 데이터들을 상황 인식에 적용할 수 있는 표현과 형태로 처리하는 모델링 단계가 필요하다. 이 두 개의 처리 단계로 요약하면 상황 정보 추출과 상황정보 모델링 단계라 말할 수 있다. 각 단계는 상황정보를 활용하는 모든 응용에서 필수적인 단계이다. 이는 상황데이터 없이 상황정보의 활용은 있을 수 없기 때문이다. 상황인식 컴퓨팅에서 상황정보 추출과 모델링 이후에는 다양한 응용이 있을 수 있다. 그러나 대부분의 응용 중 대표되는 것은 추출된 상황정보를 토대로 현재 상황을 파악하는 상황인식이다. 상황인식은 임계치 비교, 추론 등과 같은 여러 기법에 의해서 가능한데 본 연구에서 지양하는 개인화를 위해서 상황인식 처리 기법이 필요하다. 따라서 본 논문에서는 u-헬스케어 시스템에서 상황정보를 획득하기 위한 상황정보 추출 기법과 상황정보 모델링 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해서 기술하고, 3장에서는 본 논문에서 제안한 상황정보 획득에 관해 기술하며, 마지막으로 4장에서 결론 및 향후 연구로 결론은 맺는다.

2. 관련연구

2.1 상황(Context)

상황의 대한 정의는 다양한 모습으로 제시되었지만, 근원적인 문제는 대부분의 사람들이 상황이 무엇인지를 무언적으로 이해하는 반면 이를 구체적으로 규명하기 어렵다는 점이다. 상황인식 컴퓨팅 분야에서 관례적으로 사용할 수 있는 특징 정의를 도출하여 이를 각 분야의 연구에서 적용하는 시도가 있었지만 대부분 성공적이지 못하였

본 연구는 산업자원부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.

다. 상황의 정의는 연구 분야와 목적에 따라 서로 다르게 정의되고 있다. 1991년 Crowley[1]는 '사용자는 목표에 의해 행동이 유도된다' 라고 가정하고, 사용자의 상황을 인식하기 위한 시스템 모델과 사용자를 관찰하는 컴포넌트들을 관리하는 시스템 모델로 구분하여 정의하였고, 1994년 Schilit[2]는 1994년 상황인식이라는 용어를 처음으로 사용하였고, 상황을 '사람과 사물의 위치, 주변의 사용자와 객체의 식별, 그리고 그러한 객체의 변화'라고 정의하였다. 그리고 가장 최근으로는 Dey[3]가 2001년 Schilit의 상황개념을 바탕으로 '사용자와 시스템 간의 상호 작용을 위해 연관된 존재물(사람, 장소, 사물 등)을 특징짓는 모든 형태의 정보'로 상황을 정의하였다.

이러한 상황은 컴퓨팅 환경의 특성에 따라 상황 종류를 컴퓨팅 상황, 사용자 상황, 물리적 상황 등으로 분류하여 사용하고 있으며, 다음과 같은 특징들을 가진다[4].

○ 순간적인 특징을 표현한다.

상황은 정적이거나 정적으로 특징화될 수 있다. 정적 상황은 사람의 생일과 같은 변하지 않는 상황이다. 수시로 변하는 상황인식 시스템의 특징 때문에 정보의 대부분은 동적이다. 동적 상황의 지속성은 매우 변하기 쉽다.

○ 상황은 불완전하다.

상황인식 시스템에서 상황은 불완전하다. 모델화한 세계의 실제 상태를 반영하지 못하게 되면, 정보는 부정확해진다. 또한 상반된 정보를 포함한다면 그 정보는 모순될 것이고, 알려지지 않은 상황의 일부 관점(aspect)이라면 불완전해진다.

○ 상황은 상호관계를 가진다.

상황들 사이에 존재하는 관계 타입들은 사람들, 그들의 장치와 통신 채널들 사이의 관계를 알 수 있도록 해준다. 상황은 어느 한 정보나 다른 부분에서 획득되는 방법에 대한 파생 규칙에 의해 관련지어진다.

2.2 상황정보 모델링

2.2.1 Key-value 모델

Key-value 모델은 상황을 모델링하기 위한 가장 간단한 데이터 구조의 모델링 방식으로 정보를 나타내고 다루기가 쉽다. Schilit은 환경변수로서 애플리케이션에 상황(위치 정보)의 값을 제공함으로써 상황을 모델링하기 위해 Key-value를 사용하였다. Key-value 모델링 방법은 분산 서비스 프레임워크에서 자주 사용된다[5].

2.2.2 마크업 스키마(markup schema) 모델

일반적인 마크업 스키마 모델링 방법은 속성과 내용을 가진 마크업 태그로 구성된 계층적 데이터 구조이다. 특

히, 마크업 태그의 내용은 일반적으로 다른 마크업 태그로 재정의(overriding)된다. 이 모델은 간단하고 유연하며 구조화 되어 있고 편재되어 있는 컴퓨팅에 적합하다. 이 방법의 상황 모델링의 대표적인 방법은 Profiles이다. 이 모델은 마크업 언어의 상위 클래스인 SGML(Standard Generic Markup Language)에 기초를 둔다.

2.2.3 객체 지향(object oriented) 모델

일반적인 객체 지향 상황 모델링 방법은 캡슐화(encapsulation), 재사용성(reusability), 상속성(inheritance)과 같은 객체 지향 방법의 주요한 이점을 사용해 상황을 추상화하여 나타내는 방법이다. 이 방법은 유비쿼터스 환경에서 상황의 동적으로 야기되는 문제들을 해결하기 위해서 만들어졌다. 상황 처리의 세부사항은 객체 레벨에서 캡슐화되고 다른 컴포넌트들에게는 감추어진다. 상황에 대한 접근은 명시된 인터페이스들만을 통하여 제공된다.

2.2.4 로직 기반(logic based) 모델

논리는 다른 표현들이나 사실들의 집합으로부터 유도되어지는 최후의 표현이나 사실의 조건들을 정의한다. 규칙들의 집합에서 이 조건들을 기술하기 위해 형식적인 시스템이 이용된다. 따라서 로직 기반 상황 모델에서는 사실, 표현, 규칙의 정형화된 표현을 사용하여 상황을 나타낸다. 상황은 사실이라는 형식으로 나타내어지고 규칙을 통해서 새로운 사실이나 표현을 추론해낸다. 로직 기반 상황 모델을 관리하는 시스템은 이런 표현을 추가, 갱신, 제거 관리와 추론의 기능을 하는 역할을 한다.

2.2.5 온톨로지 기반(ontology based) 모델

온톨로지는 개념들과 상호작용들을 쉽게 표현할 수 있다. 최근 다양한 상황인식 프레임워크에서 온톨로지 기반의 모델을 채택하고 있는 등 OWL(Web Ontology Language)에 기반한 시맨틱 웹(Semantic Web) 연구와 관련하여 활발한 연구가 되고 있는 분야이다.

3. u-헬스케어 시스템을 위한 상황정보 획득 기법

3.1 상황정보 추출 기법

상황정보는 여러 데이터들을 통해 정의되는 특징을 가지고 있다. 따라서 유효하지 않은 데이터를 포함했을 경우에는 잘못된 상황정보가 도출되는 위험을 가지고 있다. 따라서 상황인지를 위한 각 센서 데이터들의 유효성은 반드시 검사되어야 한다.

본 논문에서는 상황정보에 대한 유효성 검사를 위해 상황정보로 활용되는 맥박, 체온, 혈압의 정상치 정보를 이용하여 해당 데이터가 정상적인 범위내의 데이터인지 검사한다. 그러나 맥박, 체온, 혈압과 같은 생체 데이터들은 활동 여부와 건강 유무에 따라 너무도 다변적인 특징을 가지고 있다. 따라서 정상 데이터의 정의가 매우 힘들다.

이러한 문제를 해결하기 위해서 운동 상황과 건강 유무 중 활동 상황에 대해서만 고려하도록 하였다. 생체 상황 정보에 기반한 유효성 검사 알고리즘은 (그림 1)과 같다.

```

1 criteriavector; //상황별 생체 데이터 기준 저장
2 hp = 맥박;
3 tm = 체온;
4 bp = 혈압;
5 len = criteriavector.lenth;
6 count=0;
7 validinfo=false;
8 REPEAT count=len
9     IF criteriavector[count].hp.lowoffset < hp and
10    criteriavector[count].hp.upperoffset > hp
11    IF criteriavector[count].tm.lowoffset < tm and
12    criteriavector[count].tm.upperoffset > tm
13    IF criteriavector[count].bp.low < bp and
14    criteriavector[count].bp.high > bp {
15        validinfo=true;
16        goto OUT;
17    }
18 END REPEAT
19 OUT;
    
```

(그림 1) 상황정보 유효성 검사 알고리즘

유효성 검사 알고리즘에서 라인 1에 있는 criteriavector는 상황별 맥박, 체온, 혈압 데이터를 포함한 객체이다. 맥박은 criteriavector 내의 hp, 체온은 tm, 혈압은 bp에 각기 대응된다. 맥박과 체온에 있는 lowoffset, upperoffset은 정상치들의 허용 오차를 의미 한다. 따라서 해당 오차 범위 내에 있는 생체 데이터는 유효한 데이터로 볼 수 있다.

상황정보의 유효성 검사는 위의 알고리즘에서 라인 8~14에서 이루어진다. criteriavector에 포함된 맥박, 체온, 혈압 데이터를 순차적으로 비교하여 상황에 따라 유효한 상황정보인지 판별한다.

그러나 센서 데이터의 특징은 지속적으로 전송되어 오기 때문에 모두 저장하려 한다면 많은 저장 공간을 할애해야 하며, 저장을 위한 처리 자원을 소모해야만 한다는 것이다. 이는 상황정보 처리에 있어서 큰 문제가 아닐 수 없다. 따라서 본 논문에서는 저장할 상황정보를 선정하는데 중복성 검사를 하여 중복된 데이터는 무시하도록 했다.

중복된 데이터란 데이터 값이 동일한 것과 이전 데이터와 크게 차이 나지 않은 데이터로 볼 수 있다. 동일한 데이터를 검색하는 처리는 어렵지 않으나, 크게 차이 나지 않아 매우 유사한 값을 갖는 경우는 중복인지 아닌지 결정하기 미묘한 것이 사실이다. 이러한 결정의 어려움은 맥박, 체온, 혈압에 대한 오프셋(offset)을 활용하는 것이다. 이 오프셋은 허용오차를 의미하며, 이를 통한 중복성 검사 알고리즘은 (그림 2)에 제시되어 있다.

(그림 2)의 중복성 검사 알고리즘을 통해 중복된 상황정보로 판정되면 해당 상황정보는 폐기된다. 매우 간단한 중복성 검사지만 오차내의 데이터에 대해 중복을 판정하기 때문에 상황정보의 관리에 있어 큰 이득이 발생한다. 유효성과 중복성 검사를 통과한 상황정보는 그리고 유효

성이 확인된 데이터는 상황정보로서 선정된다. 그리고 상황정보 기록기에 전송되어 데이터 온톨로지에 저장된다.

```

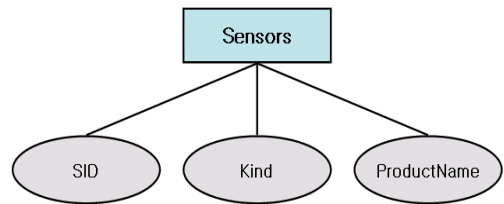
1 criteriavector; //상황별 생체 데이터 기준 저장
2 oldhp;
3 oldtm;
4 oldbp;
5 hpoffset = criteriavector[0].hp.lowoffset;
6 tmoffset = criteriavector[0].tm.lowoffset;
7 bpoffset = criteriavector[0].bp.lowoffset;
8 h_redun = false;
9 t_redun = false;
10 b_redun = false;
11 redun = false;
12 IF (hp >oldhp-hpoffset) and (hp <oldhp+hpoffset)
13     h_redun = true;
14 IF (tm >oldtm-tmoffset) and (tm <oldtm+tmoffset)
15     t_redun = true;
16 IF (bp >oldbp-bpoffset) and (bp <oldbp+bpoffset)
17     b_redun = true;
18 IF (h_redun = true) and (t_redun = true) and (b_redun = true)
19     redun = true;
    
```

(그림 2) 상황정보 중복성 검사 알고리즘

3.2 상황정보 모델링

온톨로지 기반 상황정보 모델의 구성은 센서나 컴퓨팅 자원으로부터 획득한 정보를 온톨로지 기반의 상황으로 구조화시키기 위해서 클래스와 인스턴스(instance) 개념을 사용한다. 클래스를 통해 생성된 인스턴스는 클래스에서 정의된 모든 속성들을 상속 받게 된다. 온톨로지에 정의된 클래스의 모든 속성은 자신의 인스턴스가 상속한다. 이 때 인스턴스는 실제로 센서나 컴퓨팅 자원으로부터 획득하는 데이터들을 가지게 된다. 또한 클래스는 속성(attribute)를 포함하는데, 이 속성이 실제적인 센서 데이터를 포함하게 된다. 이러한 구성 스키마를 토대로 상황정보 모델링을 위한 온톨로지를 설계하였다.

상황정보 모델은 센서, 사용자, 데이터 온톨로지로 구성하였다. 먼저 센서 온톨로지는 센서의 기본적인 정보를 모델링한 것이다. (그림 3)은 센서 온톨로지의 구성도이다.



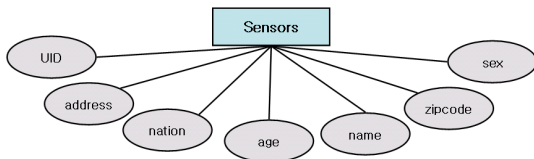
(그림3) 센서 온톨로지 구성도

센서 온톨로지는 하나의 클래스와 3개의 속성으로 구성되어 있다. 각각의 의미는 <표 1>과 같다.

<표 1> 센서 온톨로지의 구성 요소별 의미

Element	분류	의미
Sensors	클래스	센서의 세부 정보를 포함할 수 있는 정보 객체
Kind	속성	센서의 종류
ProductName	속성	센서명
SID	속성	센서 식별자

사용자 온톨로지는 사용자의 기본적인 정보를 포함하는 프로파일에 해당하는 온톨로지이다. 이 사용자 온톨로지는 (그림 4)과 같은 요소들에 의해 구성된다.

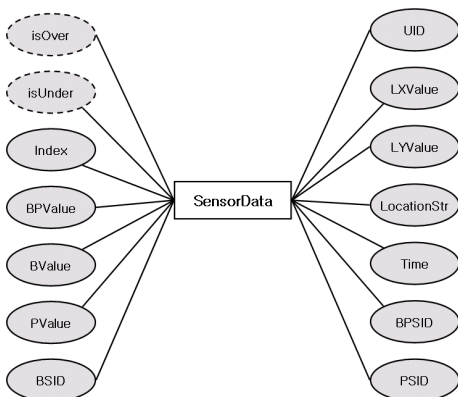


(그림 4) 사용자 온톨로지의 구성도

(그림 4)은 사용자 정보를 저장하기 위한 사용자 온톨로지의 OWL 스키마이다.

데이터 온톨로지는 센서 온톨로지와 사용자 온톨로지에 정의된 사용자 및 센서에 대한 온톨로지를 내부적으로 포함하며, 상황정보를 저장하기 위한 속성들을 포함하도록 구성된다. 데이터 온톨로지의 구성은 (그림 5)와 같다.

데이터 온톨로지는 1개의 클래스와 14개의 속성으로 구성된다. 제시된 (그림 5)에서 점선으로 표현된 isOver와 isUnder 속성을 볼 수 있다. isOver 속성은 혈압, 체온, 맥박에 해당하는 상황정보 중 상황별 정상치 이상일 때를 나타낸다. 만약 혈압에 대해 isOver가 명시되게 되면 혈압 정보가 정상치 이상이기 때문에 개인 건강의 문제가 있음을 나타낼 수 있게 된다. isUnder는 isOver의 반대 경우를 나타내는데 사용된다.



(그림 5) 데이터 온톨로지의 구성도

(그림 5)의 BSID, BPSID, PSID, UID는 온톨로지간 연관 관계에 대한 정보를 포함한다. 즉, 센싱된 데이터가 누구의 것이며, 어떤 것에 의해 측정되었는지에 대한 관계

정보를 나타내는데 사용된다는 것이다. 이러한 각 온톨로지간의 관계는 데이터 온톨로지를 중심으로 센서 온톨로지와는 “센싱”이란 관계를 가지고, 사용자 온톨로지와는 “대상자”라는 관계를 가진다. 이러한 관계와 모델링된 센싱 데이터를 통해 누군가의 건강에 대한 추론을 수행할 수 있게 되는 것이다.

4. 결론 및 향후 연구

본 논문에서는 유비쿼터스 컴퓨팅 환경에서 상황정보 획득 기법과 온톨로지 모델링을 이용한 u-헬스케어 시스템을 위한 상황정보 획득 기법을 제안하였다. 이러한 상황정보 기법은 센서로부터 수집된 데이터를 환자 및 사용자의 상황에 적합하도록 유효성 검사와 중복성 검사 알고리즘을 통하여 유효한 데이터들만을 추출할 수 있도록 하였고, 추출된 정보들을 온톨로지 기반 모델링을 통하여 정형화함으로써 상황에 대한 정보를 이용하여 실시간으로 사용자에게 u-헬스케어를 제공해줄 수 있는 추론 시스템에 적용할 수 있도록 하였다.

또한 본 논문을 통한 상황정보 획득 및 처리 기법을 u-헬스케어 시스템뿐만 아니라 유비쿼터스 컴퓨팅에서 상황정보를 이용하여 사용자에게 적절한 서비스를 제공해줄 수 있는 기타 유비쿼터스 컴퓨팅으로의 적용이 가능하다.

향후 연구로는 본 논문을 통하여 획득한 상황정보를 이용하여, 사용자 혹은 환자에게 적절한 의료 서비스를 제공할 수 있도록 추론 엔진을 이용한 의료 서비스 추천 시스템을 연구하여 적용할 것이다.

참고문헌

- [1] M. Weiser, "The Computer for the 21st Century," Scientific American, pp.94-101, September, 1991.
- [2] J.L. Crowley, J. Coutaz, G. Rey and P. Reignier. "Perceptual Components for Context Aware Computing," Proceedings of the 4th international conference on Ubiquitous Computing, Vol. 2498, pp.117-134, 2002.
- [3] A.K. Dey, "Understanding and Using Context," Personal and Ubiquitous Computing Journal, Vol. 5, No. 1, pp.4-7, 2001.
- [4] K. Henricksen, J. Indulska and A. Rakotonirainy, "Modeling context information in pervasive computing systems," in Proceedings 1st International Conference on Pervasive Computing 2414, Zurich, pp.247-261, August, 2002.
- [5] M. Samulowitz, F. Michahelles and C. Linnhoff-popien, "Capeus: An architecture for context-aware selection and execution of services." In New developments in distributed applications and interoperable systems, Kluwer Academic Publishers, pp.23-39, 17-19 September, 2001.