

유비쿼터스 컴퓨팅 환경을 위한 상황 모델 정의 및 상황 인식 프레임워크 구현

Implementation of a context-awareness framework and context model for ubiquitous computing environment

이정은, 박현정, 박두경, 윤태복, 박교현, 이지형*

JungEun Lee, HyunJung Park, DooKyung Park, TaeBok Yoon, KyoHyun Park
and JeeHyong Lee

* 성균관대학교 컴퓨터공학과

Email: {papaxi, tomean, haderme, tbyoon, megagame}@skku.edu,
jhlee@ece.skku.ac.kr

요 약

본 논문에서 제안한 시스템은 유비쿼터스 컴퓨팅 환경에서 사용자에게 상황 인식 서비스와 주변 환경에 지능적인 상호 작용을 제공한다. 따라서, 유비쿼터스 컴퓨팅의 다양한 상황과 환경에서 시스템은 컨텍스트 정보를 수집과 분석을 하기 위해 상황 인식 능력을 필요 하게 된다. 그러나 현재의 상황 인식 시스템은 다양한 타입에서 컨텍스트의 체계적인 이용에 부족하고, 단지 몇몇 시스템들은 한정된 환경에서 각자 사용자의 선호도를 획득하는 학습 매커니즘을 가진다. 본 논문에서는, 유비쿼터스 컴퓨팅 환경에서 상황 인식을 위한 개괄적인 프레임워크를 제안한다. 본 프레임워크는 다양한 센서로부터 컨텍스트의 다양한 타입을 쉽게 이용하고 생성할 수 있게 만들어졌다. 프레임워크는 동적 환경에서 사용자에게 서비스를 제공하기 위하여 센서, 통합, 추론 그리고 학습 컨텍스트 정보를 제공한다. 그리고 사용자의 건강을 실시간으로 관리할 수 있는 u-Health 시스템에 제안한 프레임워크를 적용을 하여 구현을 하였다.

Abstract

The systems in the ubiquitous computing environment need to provide users with context-aware services, intelligently interacting with the surrounding environment. Therefore, the systems in the ubiquitous computing environment require context-awareness ability in order to gather and analyze context information in various situations and environments. However, existing context-aware systems lack the ability to systematically generate and handle various types of context information, and only a few systems have ability learning from environment. In this paper, a general context model is defined to describe various contexts and a context-awareness framework is implemented based in the model, which makes it straightforward to handle and generate various types of context from diverse sensor. The framework is designed to allow a system to sensed, combined, inferred, and learned context information, in order to provide users with services in dynamic environments. We have implemented the proposed framework and applied it to a u-Health management system.

Key words : Context-awareness, Context Model, Ubiquitous Computing Environment

1. 서 론

최근 우리 생활과 밀접하게 연관되어 있는 유비쿼터스 컴퓨팅 환경에서는 컴퓨터 스스로 사용자가 처한 환경을 인식하여 적절한 서비스를 제공한다. 이를 위해서 상황 인식 기술은 필수적인데, 상황 인식을 위해서는 주변 환경으로부터 센서의 정보 수집 기술, 정보의 통합을 통한 상황 파악 및 해석, 추론 기술 등이 필요하다[8]. 그러나 이렇게 주변으로부터

터 수집되고 처리되어야 하는 컨텍스트 정보는 다양할 뿐 아니라 일반적으로 사용되는 표현 방법도 없기 때문에 현재 까지 상황 인식 기술은 특정 시스템이나 응용프로그램에 의존적인 형태로 개발되었다.

따라서 유비쿼터스 환경을 위한 시스템이나 응용프로그램을 개발할 때 이러한 중복적인 노력을 막고, 일반적으로 사용될 수 있는 상황 인식 기술과 상황 인식 프레임워크의 개발은 매우 필요하다. 이를 위해서는 다양한 타입을 갖는 컨텍스트 정보를 표현 및 정의할 수 있는 일반적인 방법이 있어야 하고, 컨텍스트가 생성되는 다양한 방법을 지원할 수 있어야 한다. 특히 하드웨어나 소프트웨어, 혹은 특정상황에 의존적이지 않은 일반적인 시스템에 사용되기 위해서는 확장 가능하고, 높은 이식성을 갖도록 상황 인식 프레임워크는 설계되어야 할 것이다. 일반적으로 환경정보 데이터는 다양한 형태로 수집 가공되어야 하므로 단순히 센서로부터 감지된

접수일자 : 2006년 5월 3일

완료일자 : 2006년 8월 14일

감사의 글 : 본 연구는 21세기 프론티어 연구개발 사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스 컴퓨팅 및 네트워크원천기반기술개발사업의 지원을 받았습니다.

정보뿐만 아니라, 이를 지능적으로 처리 가공할 수 있는 학습 등의 인공지능 기법도 적용 가능해야 한다.

본 논문에서는, 유비쿼터스 컴퓨팅 환경의 시스템에서 일반적으로 사용될 수 있고, 확장 가능한 상황 인식 프레임워크를 제안하고 개발하였다. 이를 위해서 다양한 컨텍스트를 정의하고 표현하기 위한 컨텍스트 모델을 정의하였다. 개발된 프레임워크는 환경으로부터 감지된 다양한 컨텍스트 정보를 쉽게 처리하고 생성할 수 있도록 도와주며, 다양한 컨텍스트 생성 방법(감지, 통합, 추론, 학습)을 지원하고, Java로 구현되어 다양한 플랫폼에서 특별한 가정 없이 동작할 수 있게 설계되었다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 기존 상황 인식 미들웨어의 연구를 살펴보고, 3장에서는 컨텍스트 정보를 정의하고 모델링하는 방법을 제시한다. 4장에서는 본 논문에서 제안하는 상황 인식 프레임워크에 대해 설명한다. 그리고 사용자의 건강을 실시간으로 관리할 수 있는 u-Health 시스템에 제안한 프레임워크를 적용을 하여 구현 과정을 기술하고, 마지막으로 5장에서는 결론 및 향후 연구에 대해서 설명한다.

2. 관련 연구

유비쿼터스 컴퓨팅 환경은 사용자 중심의 서비스를 제공하기 위해서 주변 환경에 대한 정보를 분석하여 응용프로그램에 제공하는 상황 인식 미들웨어가 일반적으로 존재한다 [1]. 이런 상황 인식 미들웨어는 사용자 및 주변 환경 정보를 감지하고 그 정보를 기반으로 추론하여 응용프로그램이 사용자가 필요로 하는 서비스나 정보를 제공해 줄 수 있도록 도와준다. 본 장에서는 국내외에 연구 개발된 기존의 상황 인식 미들웨어의 연구동향과 기능을 중심으로 살펴본다.

2.1 Context-Toolkit

상황 인식 미들웨어의 출발점인 Context-toolkit는 컨텍스트와 사용자의 입력 값을 분류하고 사용자, 장소, 상태, 시간 등 4개의 컨텍스트 타입으로 정의하였다[3]. 이 미들웨어의 특징은 센서로부터 취득한 컨텍스트 정보를 표현 및 통합한 후 응용프로그램에게 제공한다. 응용프로그램에게 주변 센서들로부터 수집된 정보를 제공하기 위해서, Context-Toolkit는 3개의 컴포넌트(Widgets, Aggregators, Interpreters)로 구성되어 있다. Widgets는 센서로부터 컨텍스트 정보를 수집하고 다른 컴포넌트 또는 응용프로그램간의 인터페이스를 제공해 준다. Aggregators는 Widgets으로 수집된 센서 정보를 기초로하여 컨텍스트 정보를 통합한다. 그리고 Interpreters는 하위 레벨의 컨텍스트로부터 상위 레벨의 컨텍스트를 추론하고 추론된 컨텍스트를 응용프로그램에게 제공한다.

이런 Context-Toolkit은 위의 3가지 컴포넌트에서 감지된, 통합된, 추론된 컨텍스트 정보를 처리할 수 있다고 제시했지만, 구체적인 표현 방법이나 추론 방법을 제공하지 않는다. 또한 사용자가 원하는 서비스를 제공하기 위해 필요한 학습 기법을 지원하기에는 부족하다.

2.2 Gaia

Gaia는 유비쿼터스 컴퓨팅을 지원하기 위해 제안된 분산 미들웨어로써 다양한 컴포넌트로 구성되어 있어 이기종 네트

워크 환경에서 응용프로그램을 위한 자원과 서비스를 관리해 준다[4]. 또한 동적인 환경에서 효과적으로 컨텍스트 정보를 관리하고 그 정보를 기반으로 이동 중인 사용자에게 고려한 적절한 서비스를 제공하고 효율적으로 처리할 수 있도록 다섯 개의 컴포넌트로 구성되어 있다. Event manager 컴포넌트가 변화하는 환경에 적절하게 대응할 수 있도록 이벤트 서비스 정보를 응용프로그램에게 제공한다. 그리고 Context service 컴포넌트는 사용자의 상태와 주변 상황을 고려해서 적절한 서비스를 제공한다. 또한 Presence service 컴포넌트는 응용프로그램, 디바이스, 사람, 서비스 등의 4가지 타입의 개체를 다양한 센서를 이용해서 각각의 정보를 수집하고 이용할 수 있는 자원에 대해서 정보를 갱신 한다. Space repository 컴포넌트는 센서를 통해 수집 또는 추론을 통해 생성된 컨텍스트를 저장하는 역할을 담당한다. Context file system 컴포넌트는 이동 중인 사용자를 고려한 서비스를 제공하기 위해서 디렉터리 서비스를 제공한다. File system 컴포넌트는 사용자에게 파일 이름이나 경로 대신에 가장 최근에 파일을 접근한 시간과 위치 정보를 이용해서 파일을 접근할 수 있도록 허용한다. 이처럼 Gaia는 사용자 중심적 컨텍스트 정보를 제공하기 위해서 감지된, 통합된 그리고 추론된 컨텍스트 정보를 처리한다. 그러나 사용자의 선호도와 성향을 고려한 서비스를 제공해 주기 위해서 필요한 학습 기능이 미약하다.

2.3 CAMUS

CAMUS(Context-Aware Middleware for Ubiquitous Computing Systems)는 상황 인식을 지원하기 위해 개발된 단일화된 미들웨어이다[5]. CAMUS의 주요 특징은 다른 시스템과 정형화된 지식을 재사용하고 공유할 수 있는 온톨로지 기반의 프레임워크이다. CAMUS는 다양한 센서로부터 얻은 감지된 정보, 그 정보를 기반으로 하여 통합된, 추론된 새로운 컨텍스트 정보를 생성해 낼 수 있도록 여섯 개의 컴포넌트로 구성되어 있다. 특히 더 정확한 표현력을 제공하기 위해서 다양한 추론 메커니즘을 제공한다. 이것은 다양한 타입의 로직(퍼지 로직, 서술 로직 등) 뿐만 아니라 기계 학습 기법을 사용한다. 그러나 CAMUS는 학습을 위한 컴포넌트는 명확하게 정의되거나 구조화되어 있지 않다. 위에서 언급한 바와 같이, 미들웨어는 다양한 센서로부터 획득한 센서 데이터를 통합, 추론, 학습을 통해서 다양한 컨텍스트 정보를 생성해 낼 수 있도록 제공해야 한다. 이런 컨텍스트 정보는 사용자에게 적절한 서비스를 제공하기 위해 사용될 것이다. 따라서 사용자에게 좀 더 나은 서비스를 제공하기 위해서 상황 인식 미들웨어는 감지된 컨텍스트 정보를 통합, 추론, 그리고 학습을 통해서 다양한 컨텍스트 정보를 생성하고 처리할 수 있어야 한다.

3. 컨텍스트 모델

개발된 상황 인식 프레임워크는 다양한 컨텍스트를 다루고 가공해야 한다. 컨텍스트의 원천은 응용환경과 프로그램을 사용하는 환경에 따라서 매우 다양하고, 각 응용프로그램에서 요구하는 컨텍스트 값의 타입이나 저장형태도 매우 다양하기 때문에, 상황 인식 프레임워크가 일반적인 응용프로그램에서 사용되기 위해서는 유연하고 확장 가능한 상황 모델을 가져야 한다. 이를 위해서 본 논문에서는 프레임워크의 개발에 앞서 상황 모델을 정의하였다. 정의된 상황 모델은

다음과 같은 4가지 요소로 정의된다.

Context = {Name, ValueType, Structure, Creation}

Name은 정의되는 컨텍스트의 이름이며, ValueType은 컨텍스트 값의 타입을 의미하며, Structure는 정의되는 컨텍스트가 다른 컨텍스트와 어떠한 연관관계를 갖는지 표현한 것이며, Creation은 컨텍스트가 어떠한 방식으로 생성될 것인가를 기술한 것이다. 지금까지의 대부분의 연구에서는 컨텍스트의 내용적 측면에서 구분하는 연구가 대부분이었고, 이와 같이 컨텍스트의 일반적인 표현이나 모델에 대해서는 많은 연구가 진행되지 않았다.

이를 바탕으로 본 논문에서는 컨텍스트를 표현하고 가공하였으며, 개발된 프레임워크를 사용하는 시스템에서도 컨텍스트를 위와 같이 기술만 하면 프레임워크를 이용한 상황 인식이 가능하도록 하였다. 다음에서는 위의 4가지를 본 논문에서 어떻게 구현을 하였는지 설명한다.

● 컨텍스트 이름

개발된 프레임워크에서 다루는 모든 컨텍스트가 이름을 가져야 하며, 이름은 일반적인 문자열(string)이다.

● 컨텍스트 값의 타입

어떤 컨텍스트는 어떤 타입의 값을 갖는지 사전에 정의되어야 한다. 예를 들어 방 온도를 나타내는 RoomTemp는 실수형(float)의 값을 갖게 되며, 방 안에 있는 사람의 수를 나타내는 NumPerson은 정수형(int)을 갖는다는 것이다. 본 프레임워크에서는 기본 타입으로 정수형(int), 실수형(float), 문자열(string), 논리형(boolean)을 가정하고 있다. 이외에도 다양한 타입 값을 유연하게 처리할 수 있도록 하기 위해 사용자가 타입을 정의할 수 있도록 하였으며, 사용자 정의형은 나열형(enumeration)으로 처리가 된다. 예를 들어 방안 온도를 표현하기 위해서 TempType= low, medium, high라고 정의할 수 있으며, 이것은 TempType의 컨텍스트는 low, medium, high 중의 하나의 값을 갖는다는 뜻이다.

● 컨텍스트의 구조

일반적으로 컨텍스트는 단순히 독립적으로 존재하지 않는다. 복잡하고 다양한 컨텍스트를 다루기 위해서는 관련 있는 컨텍스트를 구조화시켜서 다룰 수 있는 방법이 필요하다. 예를 들면, 어떤 회의실의 컨텍스트를 다루는 응용프로그램이 있다면, 회의실 온도(RoomTemp), 회의실의 사람 수(NumPerson), 회의실의 조도(LightIntensity) 등의 컨텍스트가 필요할 것이다. 이 세 가지는 각각이 하나의 독립된 컨텍스트이고, 회의실이라는 커다란 컨텍스트에 소속된 종속 컨텍스트이기도 하다. 본 프레임워크는 컨텍스트 사이의 이러한 종속 관계를 트리구조를 이용해서 표현할 수 있도록 하였다. 예를 들면, 위의 4가지 컨텍스트의 종속관계는 그림 1과 같이 표현할 수 있다.

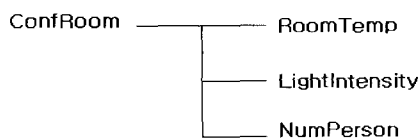


그림 1. 컨텍스트의 계층적 구조관계의 예
Fig. 1. An example of hierarchical structure of contexts

이를 위해서 본 프레임워크에서는 이러한 컨텍스트의 구조적인 정보를 따로 표현하지 않고, 컨텍스트의 이름으로써 표현하였다. 예를 들어, 앞에 말한 RoomTemp, NumPerson, LightIntensity는 모두 회의실(ConfRoom)의 종속 컨텍스트이므로, 이것들의 이름을 ConfRoom.RoomTemp, ConfRoom.NumPerson, ConfRoom.LightIntensity로 표현할 수 있게 했으며, 컨텍스트 ConfRoom의 값은 이 세개를 묶은 값으로 정의하였다. 예를 들면, ConfRoom.RoomTemp=20.0, ConfRoom.NumPerson=5, ConfRoom.LightIntensity=30 이라면 ConfRoom=(20.0, 5, 30)이 된다. 즉, 컨텍스트의 이름은 루트 컨텍스트에서 해당 컨텍스트까지의 경로를 이용해 표현한 것이며, 종말노드가 아닌 컨텍스트의 값은 그 노드의 자식들의 값의 집합으로 표현된다. 이렇게 표현함으로써, 컨텍스트는 단순히 독립적인 값으로 존재하지 않고 그 의미에 따라서 구조화 형태를 가짐으로써 컨텍스트를 쉽게 관리하고 처리할 수 있다.

● 컨텍스트의 생성타입

본 프레임워크에서는 컨텍스트가 생성되는 방식으로 4가지를 사용할 수 있게 하였다. 그것은 감지된 컨텍스트(Sensed Context), 통합된 컨텍스트(Combined Context), 추론된 컨텍스트(Inferred Context), 학습된 컨텍스트(Learned Context)이며, 각각의 설명은 다음과 같다.

감지된 컨텍스트

센서로부터 직접 얻을 수 있는 컨텍스트 값이 여기에 속한다. 유비쿼터스 환경에서 이러한 컨텍스트 정보는 센서를 통해 가장 쉽게 얻을 수 있다. 이런 컨텍스트는 사용자나 주변의 모든 물체에 부착된 다양한 형태의 센서로부터 얻어진 사용자의 체온, 위치, 빛 세기, 온도, 습도, 시간 등이다. 따라서 이러한 컨텍스트는 어떤 센서로부터 얻어지는가에 대한 추가적인 정보가 필요하다. 이런 타입의 컨텍스트를 ‘기본 컨텍스트,’ 또는 ‘비가공 컨텍스트’ 라고 부른다.

통합된 컨텍스트

기본 컨텍스트 만으로는 사용자에게 적절한 서비스를 제공하는데 필요한 여러 가지 정보를 얻기 어렵다[5]. 어떤 컨텍스트 정보는 다른 컨텍스트들의 통합을 통해 생성될 수 있다. 본 논문에서는 통합된 컨텍스트는 계산 수식을 통해 정의하였다. 예를 들어, 표준 체중은 사용자의 키로부터 산출되는 정보이다. 일반적으로 표준 체중은 사용자의 키에 100을 빼고 0.9를 곱하는 방법을 사용하여 구한다.

$$\text{StandardWeight} = (\text{Height} - 100) * 0.9$$

표준 체중과 같은 컨텍스트 정보는 감지된 컨텍스트 정보가 아니다. 따라서 이러한 타입의 컨텍스트를 정의하기 위해서 어떤 컨텍스트를 어떻게 통합할 것인가에 대한 정보가 필요하다.

추론된 컨텍스트

어떤 컨텍스트 정보는 단순히 센서나 혹은 센서 값의 단순 계산으로 얻어지기 어려운 경우가 있다. 좀 더 복잡한 방식으로 가공된 컨텍스트를 모델링하기 위하여 추론된 컨텍스트를 정의하였다. 추론된 컨텍스트는 규칙과 추론 엔진을 통하여 얻어지는 컨텍스트 정보이다. 이러한 컨텍스트에는 사용자와 주변 관계, 방안에서 사용자의 활동(미팅, 발표, 또는 TV 보기 등), 또는 사용자의 역할 등에 대한 것이 있다. 예를 들어, 사용자의 휴식 컨텍스트 정보(NeedRelaxation)는

사용자의 나이, 날씨, 사용자의 질병으로부터 추론하여 생성될 수 있으며, 이러한 컨텍스트는 다음과 같은 규칙에 의해서 생성된다면 추가적으로 생성 규칙에 대한 정보가 필요하다.

IF the user's age is more than 70 and the weather is a cold, then the user needs to relaxation.

If .User.Age >= 70 and Env.Weather <=15, Then NeedRelaxation= ture

학습된 컨텍스트

컨텍스트는 감지, 통합, 추론뿐만 아니라 학습을 통해서도 얻을 수 있다[6]. 학습된 컨텍스트는 어느 기간 동안에 생성되고 저장된 컨텍스트의 DB에 기계학습 기법이나 데이터 마이닝 기법을 적용하여 생성될 수 있다. 이러한 학습된 컨텍스트는 기존의 확실적인 서비스가 아닌 사용자의 선호도 및 성향을 고려한 서비스일 경우에 필요하다. 예를 들면, 사용자의 다음 행동, 심리상태 등 과거 사용자 데이터를 통해서 사용자가 어떤 분야에 관심이 있는지, 어떤 행동을 할지 예측하는데 사용될 수 있다.

4. 제안하는 프레임워크

본 논문에서는 그림 2와 같이 상황 인식을 위한 프레임워크를 제안하고 개발하였다. 이는 유비쿼터스 환경에서 상황 인식이 필요한 시스템에 적용될 수 있도록 확장성과 유연성을 갖도록 설계된 프레임워크이다. 제안된 프레임워크는 다양한 원천에서 생성되는 감지된, 통합된, 추론된, 그리고 학습된 컨텍스트 정보를 하드웨어나 소프트웨어에 대한 특별한 가정 없이 처리할 수 있게 설계되었다. 상황 인식 프레임워크는 Java와 Jess를 사용하여 구현되었으며, Jess는 자바 기반의 규칙기반 추론엔진으로 본 논문에서 제안한 통합된 컨텍스트와 추론된 컨텍스트를 생성하는데 사용하였다[7]. 제안한 프레임워크는 11개 모듈, 2개의 데이터베이스와 1개의 파일로 구성 되어있고, 컨텍스트 디스크립션 파일을 만들 수 있는 에디터를 개발하였다.

● 컨텍스트 디스크립션 & 초기화 모듈

그림 2의 컨텍스트 디스크립션 파일은 프레임워크에서 사용되는 모든 컨텍스트 정보를 포함하고 있다. 컨텍스트 디스크립션 에디터로 생성된 컨텍스트 디스크립션 파일은 시스템에서 필요한 컨텍스트의 구조화된 이름, 컨텍스트의 값의 타입과 관련된 수식과 규칙으로 구성된다. 디스크립션 파일은 표 1와 같이, C_name, C_type, C_value, C_rule 그리고 C_alarm으로 구성되어 있다. C_name은 컨텍스트 구조적 이름이며, 예를 들어 사용자 키를 나타내는 컨텍스트 정보는 Context.User.Biomedical.Height처럼 표시할 수 있다.

그리고 C_type은 컨텍스트 생성 타입으로 감지된(sensed), 통합된(combine), 추론된(inferred), 학습된(learned) 컨텍스트 중에 하나이다. C_type이 combined인 경우는 해당 컨텍스트를 구하기 위한 수식이 추가로 기술된다. 본 논문에서 사용한 수식은 Jess에서 수행 가능한 형식의 표현법이며, 표준 체중을 위한 수식은 다음처럼 표현된다.

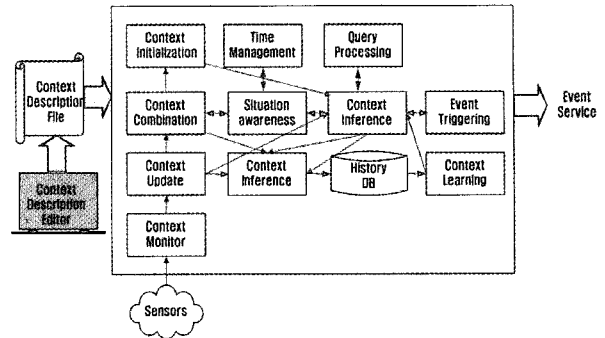


그림 2. 상황 인식을 위한 프레임워크
Fig. 2. A Framework for context-aware

표1. 컨텍스트 기술 테이블 속성

컨텍스트 속성	컨텍스트 정의
C_name	컨텍스트 이름
C_type	컨텍스트 타입
C_value	컨텍스트 값의 형 타입
C_rule	추론 규칙(rule)
C_alarm	알람 조건

(*(-Context.User.Biomedical.Height 100) 0.9)(수식1)

C_value는 해당 컨텍스트 값의 타입으로 int, float, string, 사용자 정의형, boolean 중의 하나이다. 예를 들어, Context.User.Biomedical.Height의 C_type은 본 논문에서는 float라고 정의하였다. C_rule은 추론된 컨텍스트를 생성하기 위해 필요한 규칙을 담고 있다. 추론된 컨텍스트의 생성을 위한 규칙도 Jess에서 실행 가능한 Jess 표현법이다. C_alarm은 현재의 컨텍스트와 관련된 위험한 상황이나 중요한 스케줄을 알려주기 위한 정보를 가지고 있다. 예를 들어, 일반적으로 사용자의 정상 체온은 36.5~37.5이다. 정상 체온의 범위를 넘어서 체온이 너무 높거나 체온이 너무 낮은 경우, 위험한 상황에 대한 정보를 알려주기 위한 알람 규칙은 다음과 같은 Jess 규칙으로 표현된다. 프레임워크가 처음 시작할 때, 컨텍스트 초기화 모듈은 디스크립션 파일에 정의되어 있는 모든 컨텍스트 정보와 관련 규칙을 읽어 Jess에 넣어준다. 컨텍스트 이름은 Jess의 팩트(fact)로 입력되고 C_rule이나 C_alarm은 Jess 규칙으로 입력된다. 이로써 초기화 모듈은 상황 인식 시스템이 동작하기 위한 초기화 작업을 마치게 된다.

```
(defrule rule1 (context (name
Context.User.Biomedical.BodyTemperature)
(value ?x&: (> ?x 38.0)))
=> (store Context.User.Biomedical.BodyTemperature (+
1 0 )))
```

● 컨텍스트 모니터

컨텍스트 모니터 모듈은 외부 응용프로그램과 프레임워크의 연결을 담당한다. 프레임워크는 하드웨어 센서로부터 들어온 데이터를 Jess의 팩트로 변환하여 입력하는 역할을 수행한다. 본 논문에서 제안하는 상황 인식 프레임워크를 사용하는 응용프로그램은 감지된 컨텍스트 정보를 저장하기 위해 지정된 메소드 즉, ContextMonitor()를 호출하면 된다.

ContextMonitor()를 호출하기 위해서는 사전에 특정 센서와 디스크립션 파일내의 특정 컨텍스트와 맵핑이 되어 있어야 한다. 예를 들어 어떤 센서의 값이 170.0f 이면 그 센서는 Context.User.Biomedical.Height,175.0f를 호출하면 된다. 컨텍스트 모니터 모듈은 감지된 컨텍스트 정보로 변환하여 Jess의 팩트로 입력한다. 예를 들어, ContextMonitor (Context.Uesr.Biomedical.Height,175.0f)가 호출되면, Context (Context.Uesr.Biomedical.Height ,175.0f)를 Jess의 명령어 assert를 이용해서 입력한다. 또한 디스크립션 파일을 참고 하여Context.Uesr.Biomedical.Height 컨텍스트 정보를 이용하는 통합된, 학습된 컨텍스트의 값도 갱신한다. 예를 들면, Height의 경우, 값이 변경 되었다면 이를 이용하는 통합된 컨텍스트인 StanardWeight도 같이 갱신 해줘야 한다. 이렇게 갱신된 컨텍스트 정보는 학습에 사용될 수도 있기 때문에 필터링 모듈을 거쳐 히스토리 DB에 저장된다.

● 컨텍스트 통합 모듈

컨텍스트 통합 모듈은 감지된 컨텍스트 정보를 기반으로 수식에 의해 새로운 컨텍스트 값을 계산하기 위한 역할을 담당한다. 수식은 컨텍스트 디스크립션 파일에서 가져오게 되는데, 컨텍스트 정보를 생성하기 위해 Jess를 사용하였다. 예를 들어, 표준 몸무게를 구하기 위해서는 위의 수식(1)같이 사용자의 키에 100을 빼고 0.9를 곱하는 계산과정이 필요로 하게 된다.

● 컨텍스트 추론 모듈

유비쿼터스 컴퓨팅 환경에서 사용자의 상황이나 주변 환경 정보 등 필요한 정보를 센서로부터 모두 수집할 수 없기 때문에, 기존의 컨텍스트 정보를 가공하여 더 고차원적인 컨텍스트 정보를 얻어 낼 수 있는 컨텍스트 추론 모듈이 필요하다. 추론 모듈의 동작 과정을 살펴보면, 시스템이 시작될 때 디스크립션 파일에 정의된 추론된 컨텍스트와 관련 규칙은 초기화 모듈에서 Jess에 입력이 된다. 그 후 특정 추론된 컨텍스트와 관련된 컨텍스트 정보가 변경 되었을 때, Jess는 그 컨텍스트를 규칙과 맵핑하여 새로운 정보를 생성하게 된다. 예를 들어, 2장의 휴식 컨텍스트 정보를 생성하기 위한 Jess 표현은 다음과 같다.

```
(defrule rule2 (and (context
    (name Context.User.Biomedical.Age)
    (value ?x&amp;::(> ?x70)))
    (context(nameContext.Environment.Weather)
    (value ?x&amp;::(<= ?x 15))))
=>(assert (context(name Context.User.Biomedical.NeedRelaxation)(value Yes))))
```

Jess의 규칙을 살펴보면, 두 개의 조건, Context.User.Biomedical.Age와 Context.Environment.Weather이 동시에 만족될 때, => 이하의 부분이 실행된다. 위의 Jess 규칙은 사용자의 나이가 많고 날씨가 추운 경우에, 사용자에게 휴식이 필요하다는 컨텍스트 정보를 생성한다. 즉, 기존의 모든 컨텍스트 정보를 기반으로 새로운 컨텍스트 정보를 생성하기 위해 지식기반의 전문가 지식을 사용한다. 예를 들면, 위치 정보, 사용 중인 응용프로그램, 사람의 수 등의 하위 레벨의 컨텍스트 정보를 기반으로 하여 방안에 사용자의 활동 정보, 프리젠테이션, 회의 또는 취침 등 상위 레벨의 컨텍스트 정보를 추론할 수 있다.

● 필터링 모듈

필터링 모듈은 유용한 컨텍스트만을 선택하여 히스토리 데이터베이스에 저장한다. 이 모듈은 컨텍스트 통합 모듈에서 제공받은 컨텍스트 정보 중 중복되거나 불필요한 데이터를 제거한 뒤 학습모듈에서 필요한 컨텍스트 정보를 히스토리 데이터베이스에 저장한다. 필터링 모듈은 히스토리 데이터베이스에 저장하기 전에, 사용자와 관련된 또는 응용프로그램과 관련된 컨텍스트를 그룹핑 또는 범주화 시킨다. 학습 데이터양이 너무 많거나 불필요한 데이터를 포함하면, 학습 시간도 오래 걸리고 학습 결과도 좋지 않다. 그래서 필터링 모듈을 통해서 학습에 필요한 컨텍스트 정보만을 얻는다. 어떤 컨텍스트를 위해서 어떤 데이터를 저장할 것인지를 디스크립션 파일에서 학습된 컨텍스트 부분을 보면 알 수 있다.

● 컨텍스트 학습 모듈

유비쿼터스 컴퓨팅 환경은 다양하게 변화하는 상황에 따라서 각각 사용자에게 서비스를 해줘야 한다. 그래서 유비쿼터스 컴퓨팅 환경은 이동 중인 사용자 또는 변화하는 주변 환경을 고려해서 사용자에게 좀 더 나은 서비스를 제공하기 위해서 사용자의 선호도 및 행동 패턴을 파악할 필요가 있다. 그러나 추론 시스템은 전문가에 의해 사전에 정확하게 정의된 규칙에 의해서만 동작한다는 단점이 있다. 따라서 그것은 변화하는 환경에 적절하게 대응할 수 없고 유동적 대처하기도 어렵다. 그래서 상황 인식시스템이 동적으로 변화하는 환경에서 사용자에게 적절한 서비스를 제공하기 위해서 주변의 상황을 적절히 학습할 수 있는 기능이 필요하다. 학습된 컨텍스트 정보는 서비스 예측과 준비를 위해 사용될 수 있다. 이 모듈은 히스토리 데이터베이스의 데이터에 가장 적합한 학습 방법, 즉 신경망, 결정 트리, 연관 규칙 등을 적용시켜 학습시킬 수 있다. 본 논문에 적용된 방식은 시간에 따라 변화하는 데이터의 학습에 적합한 시간 가중치를 이용한 결정트리를 이용하였다[7].

● 이벤트 트리거링 모듈

유비쿼터스 환경에서 상황 인식시스템은 현재 컨텍스트 정보를 기반으로 하여 상황에 따라 적절하게 대처 할 필요성이 있다[4]. 이벤트 트리거링 모듈은 사용자가 변화하는 주변 환경에 적절하게 대응할 수 있도록, 어떤 컨텍스트가 사전에 설정한 특정 조건을 만족시키면, 특정 이벤트를 발생시킬 수 있는 기능을 제공한다. 예를 들어, 환자를 관찰하는 시스템의 경우, 감지된 컨텍스트 정보인 Context.User.Biomedical.BodyTemperature가 38도 이상인 경우에 보호자나 간호사에게 메시지나 음성을 통해 알려줘서 적절하게 대처할 수 있도록 해야 한다. 본 논문에서 개발한 프레임워크에서는 컨텍스트 디스크립션 파일에 컨텍스트마다 조건을 설정 할 수 있다. 설정된 조건은 시스템 구동시 Jess의 규칙으로 입력되고, 수행 중 이벤트 조건이 만족되면 해당 이벤트가 발생하여 본 프레임워크를 사용하는 상황 인식 시스템에 전달되어 적절히 대응할 수 있게 한다.

● 컨텍스트 디스크립션 에디터

본 연구에서 제안된 프레임워크는 컨텍스트 디스크립션 파일만 생성을 하면 쉽게 사용할 수 있도록 설계 되었다. 이러한 컨텍스트 디스크립션 파일을 만들기 위해서는 Jess의

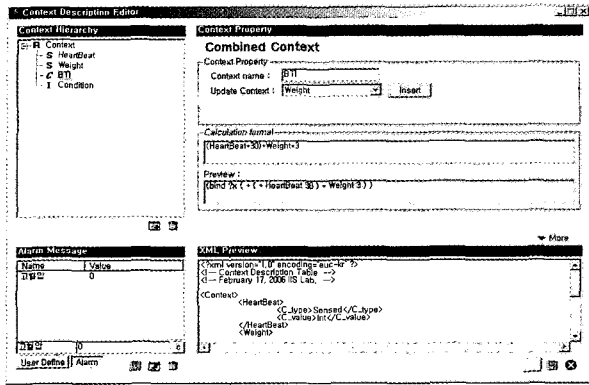


그림 3. 컨텍스트 디스크립션 에디터
Fig. 3. Context Description Editor

를과 표현에 대해서 알아야 하고 컨텍스트간의 상호 의존성과 컨텍스트 값 타입 등을 오류 없이 정확히 기술 해야 한다. 따라서, 컨텍스트 디스크립션 파일 생성시 생길 수 있는 오류를 방지하고 사용자의 편의를 위해 본 연구에서는 컨텍스트 디스크립션 파일을 쉽게 만들 수 있고 사용할 수 있도록 에디터를 개발하였다. 에디터의 구성은 컨텍스트의 계층적 구조를 표현하는 부분과 컨텍스트 속성을 생성하는 부분으로 구성되어 있다. 에디터에서 제공하는 기본 타입으로 정수형, 실수형, 문자열, 논리형을 가정하고 있다. 이외의 다양한 타입의 값을 유연하게 처리할 수 있도록 하기 위해서 사용자가 타입을 정의할 수 있도록 하였으며, 사용자 정의형은 나열형으로 처리가 된다. 또한 xml문서의 파싱 기능과 Jess의 룰과 표현에 대한 여러 체킹 기능을 제공한다.

● 구동 메커니즘

본 논문에서 구현하고자 하는 u-Health 관리 시스템의 전체 구성도는 그림 4와 같다. 시스템은 스마트 홈에서 다양한 센서들로부터 컨텍스트 정보를 센싱 한다고 가정한다. 제안한 프레임워크는 누구나 쉽게 사용할 수 있도록 컨텍스트 디스크립션 에디터를 이용하여 디스크립션 파일을 만든다. 디스크립션 파일은 제안된 프레임워크에서 사용될 모든 컨텍스트 정보를 포함하고 있으며, 컨텍스트 검증에서 사전에 미리 정의된 디스크립션 파일을 기반으로 컨텍스트 이름, 센싱된 값의 타입을 검증한다. 검증된 컨텍스트는 컨텍스트 매니저에게 제공된다. 컨텍스트 매니저는 센서된 컨텍스트 정보를 처리하고 나머지 추론, 통합, 학습된 컨텍스트 정보를 생성한다. 그리고 컨텍스트 정보를 저장하기 위해서 데이터베이스를 사용하지 않고 추론 엔진인 Jess에 저장한다. 우리 시스템은 새로운 데이터, 즉 새로운 컨텍스트 정보가 들어올 때마다 Jess에 새로운 사실로 저장된다. 또한, Jess는 새로운

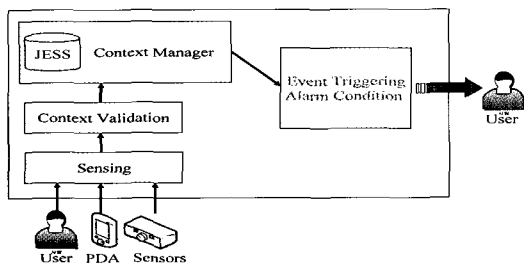


그림 4. u-Health 관리 시스템 설계도
Fig. 4. u-Health management system design

데이터가 들어올 때, 추론을 통해 새로운 컨텍스트를 생성해 낸다. Jess에 들어있는 컨텍스트 데이터를 기반으로 사용자의 건강상태를 체크하고 알람조건이나 추론에 따라 사용자의 건강 위험요인이 발생하면 이벤트 트리거링 모듈에서 위험요인 경고를 나타내며 사용자에게 위험요인을 알려준다.

5. 결론 및 향후 연구

본 논문에서는 유비쿼터스 컴퓨팅 환경의 상황 인식 시스템을 위한 프레임워크를 제안하고 개발하였다. 제안한 프레임워크는 특정 하드웨어나 소프트웨어 환경에 의존적이지 않고 일반적인 시스템에서 사용가능 하도록 확장성과 유연성을 고려하여 설계하였다. 이를 위해서 먼저 컨텍스트 모델을 정의하였으며 이를 바탕으로 시스템에서 사용될 컨텍스트 정보를 기술한 디스크립션 파일에 정의하도록 하였다. 구현은 Java와 Jess를 이용하였으며, 디스크립션 파일의 생성을 위한 에디터도 개발하였다. 또한, 제안한 프레임워크를 u-Health 관리 시스템에 적용을 하여 구현하였다. u-Health 관리 시스템에 필요한 컨텍스트 정보는 주변의 센서들로부터 사용자의 컨텍스트 정보와 주변 환경 컨텍스트 정보를 제공 받는다. 이러한 컨텍스트 정보를 바탕으로 사용자의 건강을 관리하고 위험 요인이 감지된 경우, 경고 메시지를 보내며 위험요인을 알려주는 시스템에 적용이 가능하였다. 향후 연구로, 제안한 프레임워크를 실제 유비쿼터스 컴퓨팅 환경에서 사용자에게 적합하고 효율적인 서비스를 제공할 수 있도록 시스템을 확장하고자 한다.

참 고 문 헌

- [1] A. Ranganathan, R. H. Campbell.: A middleware for Context-Aware Agents in ubiquitous Computing Environments, International Conference EUC, Vol. 3207, pp. 672-681, 2004
- [2] A.K.Dey, "Context-Aware Computing: The CyberDesk Project."AAAI, pp.51-54, 1998
- [3] A. K. Dey, G. D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," Human Factors in Computing Systems, pp. 434-441, 1999
- [4] Roman, M, "Gaia: A Middleware Infrastructure to Enable Active Spaces", IEEE Pervasive Computing, pp. 74-83, 2002
- [5] A. Shehzad, Hung Q. Ngo, Kim Anh Pham, and Sungyoung Lee, "Formal Modeling in Context Aware Systems", MRC2004, pp.13-24, 2004.
- [6] Y. T. Park, "Contexts in Ubiquitous Computing," workshop on ubiquitous computing, pp.151-172, 2004.
- [7] 동립권, 이지형, "시간 가중치 엔트로피를 이용한 결정트리 생성 알고리즘", 한국정보과학회 춘계학술대회, Vol. 32, pp. 613-615, 2005
- [8] H.Q. Ngo, A. Shehzad, S. Liaquat, M. Raiz, S. Lee, "Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework," EUC2004, Vol. 3027, pp.675-681, 2004

저 자 소 개



이정은(JungEun Lee)

2003년: 강남대학교 컴퓨터공학과(학사)
2005년~현재 : 성균관대학교 컴퓨터공학 석사과정

관심분야 : 사용자 모델링, 상황 인식, 학습/추론시스템
Phone : +82-31-290-7987
E-mail : papaxi@skku.edu



박현정(HyunJung Park)

2004년: 단국대학교 산업공학과(학사)
2006년: 성균관대학교 컴퓨터공학과 (공학석사)

관심분야 : 상황 인식, 유비쿼터스 컴퓨팅
Phone : +82-31-290-7987
E-mail : tomen@skku.edu



박두경(DooKyong Park)

2005년 : 성균관대학교 컴퓨터공학(학사)
2006년~현재 : 성균관대학교 컴퓨터공학 석사과정

관심분야 : 멀티에이전트, 상황 인식
Phone : +82-31-290-7987
E-mail : haderme@skku.edu



윤태복(KaeBok Yoon)

2001년 : 공주대학교 컴퓨터공학(학사)
2005년 : 성균관대학교 컴퓨터공학(석사)
2005년~현재 : 성균관대학교 컴퓨터공학 박사과정

관심분야 : 게임AI, 멀티에이전트
Phone : +82-31-290-7987
E-mail : tbyoon@skku.edu



박교현(KyoHyun Park)

2006년 : 광운대학교 컴퓨터공학(학사)
2006년~현재 : 성균관대학교 컴퓨터공학 석사과정

관심분야 : 게임AI, 멀티에이전트
Phone : +82-31-290-7987
E-mail : megagame@skku.edu



이지형(JeeHyong Lee)

1993년 : 한국과학기술원 전산학과 공학사
1995년 : 한국과학기술원 전산학과 공학석사
1999년 : 한국과학기술원 전산학과 공학박사
2000년 : 한국과학기술원 첨단정보기술 연구센터 위촉 연구원

2002년 : 미국 SRI International, International Fellow
2002년~현재 : 성균관대학교 정보통신공학부 조교수

관심분야 : 지능시스템, 기계학습, 온톨로지
Phone : +82-31-290-7154
E-mail : jhlee@ece.skku.ac.kr