

유비쿼터스 컴퓨팅 미들웨어를 위한 상황 인식 프레임워크

박현정^o 이지형성균관대학교 정보통신공학부 컴퓨터공학과
tomean@skku.edu, jhlee@ece.skku.edu

A framework of context-awareness for ubiquitous computing middlewares

Hyun-Jung Park^o Jee-Hyong Lee

Department of Computer Engineering, Sungkyunkwan University

요 약

오늘날 모든 물체에 컴퓨터가 내장되어 언제 어디서나 사용자에게 적절한 서비스를 제공하는 유비쿼터스 컴퓨팅 개념이 대두되고 있다. 그래서 유비쿼터스 컴퓨팅 환경은 사용자 중심의 환경이나 주변 상황으로부터 획득한 컨텍스트를 분석하기 위해서 상황 인식(context-awareness) 시스템이 필요하다. 이 시스템은 센서로부터 취득한 컨텍스트, 즉 하위 레벨의 컨텍스트로부터 통합, 추론, 그리고 학습을 통해서 상위 레벨의 컨텍스트를 생성해야 한다. 그러나 기존의 상황 인식 시스템은 다양한 타입의 처리할 수 없고 사용자의 선호도나 흥미도를 고려하지 않은 제한점이 있다. 본 논문에서는 유비쿼터스 컴퓨팅 미들웨어를 위한 다양한 센서들로부터 다양한 타입의 컨텍스트를 처리할 수 있는 생성할 수 있는 프레임워크를 제안한다. 이 프레임워크는 변화하는 주변 환경에 따라 사용자에게 다양한 서비스를 제공할 수 있도록 센서, 통합된, 추론된, 그리고 학습된 컨텍스트를 생성할 수 있도록 설계하였다.

1. 서 론

점차 컴퓨팅의 동향은 모든 물체에 컴퓨터를 내장하여 이기종 컴퓨터간의 네트워크로 연결하여 상호간에 협조와 타협을 해 가면서 언제 어디서나 눈에 보이지 않게 사용자에게 적절한 서비스를 제공해 주는 유비쿼터스 컴퓨팅 환경으로 변화하고 있다[1]. 이러한 유비쿼터스 환경에서는 사용자에게 제공하는 서비스에 대한 요구를 만족시키기 위해서 상황 인식(context-awareness) 시스템이 필요하다. 유비쿼터스 환경에서 사용자에게 서비스 또는 정보를 제공해 주기 위해서는 상황 인식 시스템(context-aware system)은 주변 환경으로부터 컨텍스트(context) 정보를 수집하고 분석하는 작업이 필요하다. 또한 상황 인식 시스템은 다양한 센서들로부터 획득한 컨텍스트(context)를 기반으로 가공처리 하여 사용자에게 적절한 서비스를 제공해야한다[3]. 그리고 이 기술은 유비쿼터스 환경에서 끊임없이 변화하는 주변 환경에 적용하기 위해서 하위 레벨 컨텍스트(context)로부터 통합, 추론, 학습 등을 통해서 상위 레벨 컨텍스트(context)를 생성할 필요성이 있다. 그래서 많은 수의 상황 인식 시스템이 유비쿼터스 컴퓨팅 환경을 지원하기 위한 목적으로 발전되었다[2].

그러나 이들 시스템들의 대부분은 이동 중인 사용자나 또는 각 사용자의 선호도 및 사용자의 흥미도를 고려하지 않는 제한점을 가지고 있다. 그리고 유비쿼터스 컴퓨팅 환경은 사용자에게 끊임없이 변화하는 환경, 정보를 얻기 어려운 주변 여건 등을 고려해서 다양한 서비스를 제공해야 한다. 그러므로 상황 인식 시스템은 동적인 환경에서 사용자를 지원 할 수 있도록 다양한 타입의 컨텍스트를 처리하고 생성할 수 있어야 한다. 본 논문에서 제안한 프레임워크를 통해서 센서된(sensed), 통합된(combined), 추론된(inferred) 그리고 학습된(learned) 컨텍스트를 생성해 내겠다.

본 논문에서는, 상황 인식 미들웨어에서 다양한 컨텍스트(context)를 처리할 수 있고 생성할 수 있는 프레임워크를 제안한다. 제안된 프레임워크는 다양한 타입의 컨텍스트(context)를 센서된 데이터로부터 통합, 추론, 그리고 학습을 통해서 쉽게 생성할 수 있다. 그래서 이를 바탕으로 사용자에게 적절한 서비스를 제공하는데 사용 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 컨텍스트의

정의와 기존의 상황 인식 미들웨어 관련연구에 대해 살펴보고, 3장과 4장에서는 유비쿼터스 컴퓨팅 미들웨어를 위한 상황 인식 프레임워크와 구현에 대해 설명한다. 마지막으로 5장에서는 결론과 향후 연구로 본 논문을 맺는다.

2. 관련 연구

본 장에서는 상황 인식 애플리케이션에서, 중요한 요소인 컨텍스트의 정의와 이 컨텍스트를 이용하여 사용자에게 서비스를 제공하는 기존의 상황 인식 미들웨어를 살펴본다.

2.1 컨텍스트의 정의

유비쿼터스 컴퓨팅 환경에서 상황 인식 시스템의 주요 목적은 사용자의 현재 상황이나 디바이스로부터 수집된 컨텍스트 정보를 이용하여 사용자가 원하는 서비스를 미리 예측하여 사용자에게 가장 적절한 서비스를 제공하는 것이다[4]. 이처럼 컨텍스트는 유비쿼터스 컴퓨팅 환경에서의 지능형 애플리케이션에서 고려해야할 중요한 요소이다. 컨텍스트는 쓰이는 애플리케이션마다 다르게 정의되지만, 일반적으로 유비쿼터스 환경에서 존재하는 개체들(사람, 장소, 객체 등)의 상황을 특성화시키기 위해 사용되는 정보로 정의된다. 좀 더 구체적으로 사용자와 애플리케이션 사이에 관련된 정보, 상황 정보, 그들 사이의 상호 관계 정보 등으로 정의할 수 있다. 이런 컨텍스트를 파악하는 방식에 따라 다음과 같이 정의할 수 있다.

● **센서된 컨텍스트(sensed context)**: 유비쿼터스 환경에서 센서를 통해서 컨텍스트를 가장 쉽게 얻을 수 있다. 이런 컨텍스트는 다양한 형태의 센서(예를 들어, 사용자 ID, 빛 센서, 온도, 위치, 습도, 시간 등) 또는 사용자에게 부착된 센서나 주변 모든 물체에 부착된 센서로부터 얻는다. 이런 타입의 컨텍스트를 '기본의 컨텍스트(elementary context)', 또는 '가공되지 않는 컨텍스트(raw context)'라고 부른다.

● **통합된 컨텍스트(combined context)**: 기본적인 컨텍스트(elementary context)만으로는 사용자에게 적절한 서비스를 제공할 수 없다. 그래서 사용자의 상태, 주변 상황, 이용할 수 있는 주변의 자원 등을 복합적으로 고려해야한다. 상위 레벨의 컨텍스트는 기본의 컨텍스트로부터 생성해 낼 수 있

다. 하지만 센서된 컨텍스트는 비구조화된 정보이기 때문에 간단한 변환이나 센서된 데이터의 표준화된 표현이 필요하다. 즉, 혼합하게 흩어진 정보들을 결합하여 구조화 시키는 변환작업이 요구된다. 센서된 컨텍스트들의 결합 또는 통합된 컨텍스트들의 결합들로부터 생성된 컨텍스트는 확실한 구조 안에서 재표현된다. 예를 들어, Context-Toolkit 에서의 구성요소인 Aggregator가 센서를 통해 얻은 기본적인 컨텍스트를 통합, 취합하여 복합적인 컨텍스트를 제공한다.

- 추론된 컨텍스트(inferred context) : 우리는 센서로부터 수집되거나 또는 통합된 컨텍스트를 기반으로 하여 많은 상위 레벨의 컨텍스트를 추론해 낼 수 있다. 또한 사용자와 주변 관계, 방안에서 사용자의 활동(예를 들어, 미팅, 발표, 또는 TV 보기 등), 또는 사용자의 역할에 대해서 추론할 수 있다. 상위 레벨의 컨텍스트를 추론하기 위해서 약간의 도메인 지식이 필요하다. 이런 도메인 지식은 센서된 또는 통합된 컨텍스트 정보를 추론하는데 도움이 될 것이다. 추론된 컨텍스트는 센서된 또는 결합된 컨텍스트들이 아닌 새로운 정보이다.
- 학습된 컨텍스트(learned context) : 컨텍스트는 센싱, 추론 뿐만 아니라 학습을 통해서도 얻을 수 있다[5]. 기존의 획일적인 서비스가 아닌 사용자의 선호도 및 성향을 고려한 서비스일 경우, 학습 기법이 적절히 요구된다. 수많은 데이터에서 추론을 통해서 사용자의 패턴이나 선호도를 파악하기는 어렵기 때문에 학습이나 데이터 마이닝 기법을 통해서 알아낼 수 있다. 예를 들면, 사용자의 행동, 말, 그리고 심리상태 등 과거 사용자 데이터를 통해서 사용자가 어떤 분야에 관심이 있는지, 어떤 행동을 할지 예측할 수 있다.

2.2 상황 인식 미들웨어

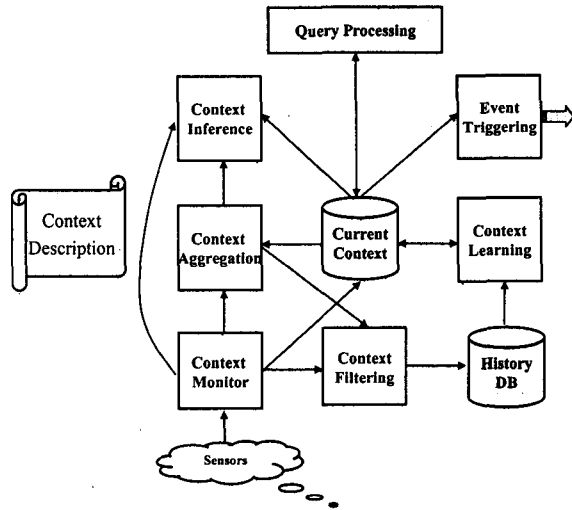
유비쿼터스 컴퓨팅 환경에서 사용자 중심적 서비스를 제공하기 위해서는 사용자의 현재 상황과 주변 상황 정보를 이용하여 사용자가 원하는 서비스가 무엇인지 파악하는 역할을 담당하는 것이 상황 인식 애플리케이션이다[3]. 다양한 정보를 수집하는 센서와 사용자에게 적절한 서비스를 제공하는 애플리케이션 사이에 중간 매개체 역할을 하는 상황 인식 미들웨어가 일반적으로 존재한다. 이런 상황 인식 미들웨어는 사용자 및 주변 환경 정보를 감지하고 그 정보를 기반으로 추론하여 애플리케이션이 사용자에 원하는 필요로 하는 서비스나 정보를 제공해 줄 수 있도록 도와준다. 본 장에서는 국내외에서 기존에 연구 개발된 상황 인식을 위한 미들웨어 Context-Toolkit[2]에 대해 살펴볼 것이다. 위의 4개의 미들웨어 중 상황 인식 미들웨어의 출발점인 Context-toolkit는 컨텍스트와 사용자의 입력 값을 분류하고 사용자, 장소, 상태, 시간 등 4개의 컨텍스트 타입으로 정의하였다. 이 미들웨어 특징은 센서로부터 취득한 컨텍스트 정보를 표현 및 통합을 한 후 애플리케이션에게 제공한다[2]. 애플리케이션에게 주변 센서들로부터 수집된 정보를 제공하기 위해서, Context-Toolkit은 3개의 컴포넌트(Widgets, Aggregators, Interpreters)로 구성되어 있다. Widgets는 센서로부터 컨텍스트 정보를 수집하고 다른 컴포넌트 또는 애플리케이션간의 인터페이스를 제공해준다. Aggregators는 widgets으로 수집된 센서 정보를 기반으로 하여 컨텍스트 정보를 통합한다. 그리고 interpreters는 하위 레벨의 컨텍스트로부터 상위 레벨의 컨텍스트를 추론하고 추론된 컨텍스트를 애플리케이션에게 제공한다. 이런 Context-Toolkit은 위의 3가지 컴포넌트에서 센서된(sensed), 통합된(combined) 그리고 추론된(inferred) 컨텍스트 정보를 처리할 수 있다고 제시했지만, 구체적인 표현방법이나 추론방법을 제공하지 않는다. 또한 사용자를 만족시킬 만한 서비스, 즉 사용자의 선호도나 흥미도를 고려한 서비스를 제공하기 위한 필요한 학습(learning) 기법을 지원하기에 부족하다.

이처럼 미들웨어는 다양한 센서로부터 획득한 센서 데이터를 통합, 추론, 학습을 통해서 다양한 컨텍스트 정보를 생성해 낼 수 있도록 제공해야 한다. 이런 컨텍스트 데이터는 사용자에게 적절한 서비스를 제공하기 위해 사용될 것이다. 사용자에게 좀

더 나은 서비스를 제공하기 위해서 상황 인식 미들웨어는 센서된 컨텍스트 정보를 통합, 추론, 그리고 학습을 통해서 다양한 컨텍스트 정보를 생성하고 처리할 수 있어야 한다.

3. 제안된 프레임워크

본 논문에서는 상황 인식 시스템을 제공하기 위해 새로운 프레임워크를 개발하였다. 이는 상황 인식을 위한 일반적인 프레임워크이다. [그림 1]은 센서된, 통합된, 추론된, 그리고 학습된 컨텍스트 정보를 하드웨어에서 미들웨어에서 특별한 가정 없이 동작할 수 있게 설계되었다.



[그림 1] : 상황 인식을 위한 프레임워크

센서로부터 획득한 컨텍스트를 사용하여 사용자에게 적절한 서비스를 제공하기 위해서는, 프레임워크는 다양한 센서들과 에이전트(Agent)로 부터 수집한 컨텍스트를 통합, 추론, 그리고 학습 기능을 포함해야 한다. 그림 1에서 보여준 프레임워크는 8개 모듈과 2개의 데이터베이스로 구성되어 있다. 각각의 모듈에 대해서 간단히 설명하겠다.

Context Description Module

Context description 모듈은 센서로부터 획득한 컨텍스트 정보를 관리한다. 일반적으로 상황 인식 애플리케이션(context-aware application)은 유비쿼터스 환경에서 다양한 센서로부터 획득한 컨텍스트 정보를 완전하고 정확하게 가정하고 사용을 한다. 그러나 실제 센서된 컨텍스트 정보는 소음(noise)이나 센서 실패(sensor failure) 또는 사용자의 실수(user error)로 인해서 이용할 수 없는 경우가 많다[6]. 이런 불확실하고 애매모호한 컨텍스트 정보를 관리하기 위해서 XML를 이용한다. 또한 센서들로부터 들어온 컨텍스트를 저장하기 전, 사전에 미리 정의된 context description table를 이용해서 센서된 컨텍스트의 유효성을 검증한다. 이를 위해 파서(parser)를 이용해서 xml 인스턴스를 검증한다.

Context Normalization Module

Context normalization module는 센서들로부터 들어온 컨텍스트 정보를 정규화된 구조로 변환시켜주는 역할을 담당한다. 유비쿼터스 컴퓨팅 환경에서는 일반적으로 다양한 센서들로부터 정보를 수집한다. 그러나 센서 데이터들의 다양한 타입, 예를 들어, 바이너리(binary), 정수(integer), 실수(real number), 문자(character), 숫자와 문자들로 조합된

데이터 등 존재한다. 이런 각각의 센서된 데이터는 표현방법이 다양하기 때문에, 컨텍스트 정보를 사용하는데 어려움이 있을 뿐만 아니라 하위 레벨의 컨텍스트로부터 상위 레벨의 컨텍스트를 생성할 때 어려움이 있을 것이다. 이 모듈은 센서된 데이터를 정규화 된 구조를 이용해서 나타낼 것이다. 또한 Context normalization module는 데이터 추상적(data abstraction) 개념을 명확하게 표현하고 데이터 변환(data conversion), 즉 화씨의 온도를 섭씨로 바꿔주는 역할을 담당한다.

Context Aggregation Module

context aggregation module는 context normalization module로부터 제공받은 컨텍스트 정보를 센서된 컨텍스트를 기반으로 하여 통합된 컨텍스트를 생성한다. 이 모듈은 사용자 ID, 디바이스 ID, 또는 위치 ID를 기반으로 하여 센서된 컨텍스트를 구조화된 형태로 통합한다. 이 구조체는 객체(object-oriented) 또는 프리딕케이트(predicate-based) 기반으로 하여 사용되었다. 이 모듈의 주요 목적은 흩어진 컨텍스트 정보를 사용할 수 있도록 잘 구조화된 구조체로서 통합을 하는 것이다. 이 모듈은 통합하기 위해서 도메인 또는 물에 대한 지식이 필요하다. 통합된 컨텍스트는 public 뿐만 아니라 individual에게 좀 더 나은 서비스를 제공하기 위해서 사용될 것이다. 이 모듈의 결과물은 현재 컨텍스트(current context)뿐만 아니라 필터링 모듈을 통해서 데이터베이스에 저장된다.

Context Inference Module

센서들로부터 모든 정보를 수집할 수 없기 때문에, context inference module이 필요하다. context inference module는 하위 레벨의 컨텍스트 정보로부터 상위 레벨의 컨텍스트 정보를 생성해낸다. 모듈은 현재의 컨텍스트 정보로부터 새로운 컨텍스트 정보를 추론해 낼 수 있다. 새로운 컨텍스트 정보는 사용자 상황이나 주변 환경이 변화할때마다 추론이 될 것이다. 또한 애플리케이션이 query processign 모듈에게 컨텍스트 정보를 요청했을 때, 그 정보가 존재하지 않는 경우에는 추론을 통해서 애플리케이션이 원하는 새로운 정보를 생성해 낸다. 우리는 새로운 컨텍스트 정보의 생성해 내기 위해서 지식기반(knowledge base)의 전문가 지식을 사용한다.

Filtering Module

Filtering 모듈은 유용한 컨텍스트만을 선택하여 히스토리 데이터베이스에 저장한다. 이 모듈은 context normalization 과 context aggregation 모듈로 제공받은 컨텍스트 정보의 중복성 또는 불필요한 데이터를 제거한다. 그래서 Learning 모듈에서 사용자 관련 컨텍스트 정보를 학습하기 위해 필요한 컨텍스트 정보를 히스토리 데이터베이스(history DB)에 저장한다. 이 모듈은 히스토리 데이터베이스(history DB)에 저장하기 전에, 사용자와 관련된 또는 애플리케이션과 관련된 컨텍스트를 그룹핑(grouping) 또는 카테고리(category) 화시킨다.

Context Learning Module

유비쿼터스 컴퓨팅 환경은 다양하게 변화하는 상황에 따라서 각각 사용자에게 서비스를 해주어야 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 알아야 한다. 그래서 학습된 컨텍스트 정보는 서비스 예측(predication)과 준비(preparation)를 위해 사용될 수 있다. 그러나 추론 시스템은 전문가에 의해 룰(rule)을 정확하게 정의해야 한다는 단점이 있다. 또한 그것은 변화하는 환경에 적절하게 대응 할 수 없고 유동적 대처하기도 어렵다. 그래서 미들웨어는 동적으로 변화하는 환경에서 사용자에게 적절한 서비스를 제공하기 위해서 학습 메카니즘이 필요하다. 이

모듈은 히스토리 데이터베이스의 데이터에 가장 적합한 학습 방법, 즉 신경망(neural network), 결정 트리(decision tree), 연합 룰 찾기(association rule finding) 등 적용시켜서 학습시킬 수 있다. 일반적으로 학습하는데 시간이 오래 걸리고 많은 양의 데이터를 필요로 한다. 그러나 유비쿼터스 컴퓨팅 환경에서의 디바이스들은 자원이 불충분하고 빠르고 끊임없이 변화하는 컨텍스트 정보를 지속적으로 증가(incremental)하여 학습 할 필요성이 있다.

Event Triggering Module

유비쿼터스 환경에서의 지능형 미들웨어는 현재 컨텍스트 정보를 기반으로 하여 상황에 따라 적절하게 행동할 필요성이 있다[4]. Event Triggering Module는 사용자가 새로운 환경에 적절하게 대응할 수 있도록 어떤 특별한 이벤트를 제공한다. 예를 들어, 사용자 주변에서 위험한 일이 발생했을 경우에 사용자에게 메시지나 음성을 통해 알려줘서 적절하게 대체할 수 있도록 한다. 또는 중요한 업무(회의, 수업, 스케줄링 등)이나 가족의 기념일 등 중요한 정보를 알려준다.

4. 실험

우리는 음악 추천시스템을 프레임워크 기반으로 하여 구현하였다. 이 시스템은 사용자의 선호하는 음악 중 현재 분위기에 맞는 음악을 추천한다. 방안의 조명 밝기와 온도 감지하는 두 센서를 이용하여 방의 분위기를 파악한다. 그리고 사용자의 선호하는 음악을 히스토리 데이터에 쌓아 둔다. 그래서 수집된 데이터에 대한 컨텍스트 표준화, 현재 분위기를 추론 하는 컨텍스트 추론 모듈, 사용자가 선호하는 음악을 찾아내는 학습 모듈 등을 구현하였다.

5. 결론 및 향후 연구

본 논문에서는 유비쿼터스 컴퓨팅 미들웨어를 위한 상황 인식 프레임워크를 제안한다. 이 제안된 프레임워크는 동적으로 변화하는 사용자와 주변 환경을 고려해서 적절한 서비스를 제공하기 위해서 다양한 타입의 컨텍스트 즉 센서된, 통합된, 추론된, 그리고 학습된 컨텍스트를 쉽게 처리할 수 있도록 설계되었다. 또한 우리는 현재 컨텍스트의 정보에 따라서 사용자나 시스템에게 중요한 정보를 알려주기 위해 이벤트 트리거링을 제공한다. 제안된 프레임워크를 기반으로 하여 모든 구성 모듈을 구현할 예정이다.

6. 참고문헌

- [1] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, pp.94-101, 1991.
- [2] 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.
- [3] A. Ranganathan, R. H. Campbell, "A middleware for Context-Aware Agents in ubiquitous Computing Environments," *International Conference EUC 2004*, Vol. 3207, pp. 672 - 681, 2004.
- [4] H.Q. Ngo, A. Shehzad, S. Liaquat, M. Raiz, S. Lee, "Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework," *Embedded and Ubiquitous Computing, International Conference EUC 2004*
- [5] Y. T. Park "Contexts in Ubiquitous Computing," *workshop on ubiquitous computing*, pp. 151-172, 2004.
- [6] K. Henriksen, J. Indulska, "Modeling and using imperfect context information", *Pervasive Computing and Communications Workshops, the Second IEEE Annual Conference*, pp. 33 - 37, 2004