wear-UCAM: 착용형 컴퓨팅을 위한 정형화된 맥락 인식 응용 모형

(wear-UCAM: Unified Context-aware Application Model for Wearable Computing)

홍 동 표 † 우 운 택 ††
(Dongpyo Hong) (Woontack Woo)

요 약 본 논문에서는 착용형 컴퓨팅을 위한 정형화된 컨텍스트 인식 응용 모형인 wear-UCAM (Unified Context-aware Application Model for Wearable Computing)을 제안한다. 유비쿼터스 컴퓨팅에 대한 관심이 고조되고 이와 함께 관련된 기술들이 발전함에 따라서, 언제 어디서나 손쉽게 컴퓨팅 자원을 활용할 수 있는 사용자 중심의 착용형 컴퓨팅에 관한 연구도 학계나 산업계에서도 활발히 진행 중이다. 제안된 wear-UCAM은 유비쿼터스 컴퓨팅 환경에서 사용자와 관련된 개인 정보를 센서로부터 획득하고 획득된 정보를 처리 및 분석해서 사용자의 컨텍스트에따른 개인화된 서비스를 제공할 수 있는 모델이다. 제안된 wear-UCAM의 특징은 다음과 같다. 1) 센서에서 획득된 정보로부터 사용자 정보 (User Profile)의 갱신, 2) 사용자의 생체 신호 수집 및 생체 신호 분석, 그리고 3) 다른 착용형 컴퓨터나 환경으로부터 사용자에 대한 개인 정보 보호이다. 본 논문에서 제안된 wear-UCAM은 컨텍스트 처리 과정의 추상화 및 센서와 응용 서비스간의 독립성 보장을 위한 다양한 컴포넌트들을 포함하고 있다. 따라서, 제안된 wear-UCAM은 유비쿼터스 컴퓨팅 환경에서 착용형 컴퓨팅에 필요한 사용자 중심의 컨텍스트기반 어플리케이션을 지원할 수 있는 모델이다. 본 논문에서는 제안된 wear-UCAM의 설계와 디자인된 wear-UCAM의 구체적인 구현 방법에 대해서 자세히 설명한다.

키워드 : 착용형 컴퓨팅, 컨텍스트 인식 응용 모형, 유비쿼터스 컴퓨팅

Abstract In this paper we propose wear-UCAM, which is a toolkit for context-aware application model in wearable computing. As the rapid developments of mobile technologies and relevant technologies, the interests in wearable computer also become indispensible in both academic and industrial fields. However, there are few research activities on the application framework or toolkit for wearable computing. Hence, we suggest wear-UCAM as the development toolkit for wearable computing applications, where we focus on how to collect relevant user context, manage it, and provide services based on the recognized context. The proposed wear-UCAM includes the abstraction of context processing as well as independence among sensors and services with other components. For the sake of rapid prototyping of the proposed toolkit, we utilize PDA with wireless LAN as a wearable computer. The detail explanation of the implementation and its discussion are presented in this paper.

Key words : Wearable Computing, Context-aware Application Model, Ubiquitous Computing

1. 서 론

최근 유비쿼터스 컴퓨팅의 확산과 더블어 관련된 기술들이 발전함에 따라서 착용형 컴퓨팅에 관한 관심도 증대되고 있다[1]. 일반적으로 유비쿼터스 컴퓨팅 환경에서는 사용자에게 보다 나은 서비스 혹은 컴퓨팅 리소스를 제공하기 위해서 다양한 센서를 환경에 설치하고, 사용자를 추적해야하는 환경적 인프라가 요구된다[2]. 따라서, 환경에서 사용자의 행동 및 상태를 감시하고 사용자의 정보를 추출하기 때문에, 개인 정보 유출과 같은 사생활 침해 문제 야기와 개인화된 정보 관리가 쉽지 않다[3]. 하지만, 유비쿼터스 컴퓨팅 환경과 달리 착용형 컴퓨팅에서는 사용자 개인과 관련된 정보를 센서로부터 수집하고 분석하여 사용자에게 적절한 서비스를 제공할

† 학생회원 : 광주과학기술원 정보통신공학과
dhong@gist.ac.kr
†† 종신회원 : 광주과학기술원 정보통신공학과 교수
wwoo@gist.ac.kr
논문접수 : 2005년 5월 11일
심사완료 : 2005년 10월 21일

수 있어야 한다. 즉, 개인과 관련된 정보를 환경에서 처리하는 것이 아니라 착용형 컴퓨터를 통해서 처리함으로써 유비쿼터스 컴퓨팅 환경에서 개인 정보를 보호하면서도 사용자가 필요한 서비스를 제공할 수 있어야 한다[3,4]. 착용형 컴퓨팅이 주목 받는 이유는 이와 같이 유비쿼터스 컴퓨팅 환경이 갖는 취약점을 보완할 수 있기 때문이다. 하지만, 착용형 컴퓨팅을 위한 구체적인 어플리케이션 모델에 관한 연구는 유비쿼터스 컴퓨팅과 관련된 연구에 비해서 아직까지 미비하다[5-9].

본 논문에서는 착용형 컴퓨팅을 위한 정형화된 컨텍스트 인식 응용 모형인 wear-UCAM(Unified Context-aware Application Model for Wearable Computing)을 제안한다. 제안된 wear-UCAM은 유비쿼터스 컴퓨팅 환경에서 사용자와 관련된 개인 정보를 센서로부터 획득하고 획득된 정보를 처리 및 분석해서 사용자의 컨텍스트에따른 서비스를 제공할 수 있는 모델이다. 제안된 wear-UCAM의 특징은 다음과 같다. 1) 센서로부터 센싱된 정보로부터 사용자 정보(User Profile)의 갱신, 2) 사용자의 생체 신호 수집 및 생체 신호 분석, 그리고 3) 다른 착용형 컴퓨터나 환경으로부터 사용자의 개인 정보 보호이다. 본 논문에서는 제안된 wear-UCAM에서 정의한 센서와 서비스간에 필요한 컨텍스트의 처리 과정을 추상화하여 제시한다. 이와 같은 컨텍스트의 처리 과정을 추상화하기 위해서 제안된 wear-UCAM에서는 다양한 소프트웨어 디자인 패턴(Design Pattern)을 적용하여 각 모듈간의 컨텍스트 처리 과정을 추상화하였다[10,11]. 제안된 wear-UCAM의 컴포넌트인 Context-Manager는 다양한 센서로부터 추출된 정보를 통합하고 관리하여 지속적인 사용자 정보를 갱신한다. 이와 같이, 지속적인 사용자 정보의 갱신을 통해서 사용자는 자신에게 필요한 개인화된 서비스를 제공받는 것이 가능하다[12]. 사용자의 생체 신호 수집 및 분석은 착용형 컴퓨팅에 있어서, 사용자와 가장 밀접한 정보로써 사용자 컨텍스트에 있어서 중요한 요소이다. 따라서 제안된 wear-UCAM에서는 생체 신호 추출 및 분석과 관련된 기능을 제공 한다[13]. 그리고 제안된 wear-UCAM은 wearSensor와 wearService로 센서와 서비스를 분리함으로써 다양한 센서와 응용 프로그램들 간의 독립성을 보장한다. 센서와 서비스간의 독립성 보장은 유비쿼터스 컴퓨팅 환경과 같이 분산 컴퓨팅 환경에서 필요한 요소들 중 하나이다. 또한, 개인화된 서비스를 제공받기 위해서 암묵적으로 사용자 정보를 서비스나 환경에게 전달하는 것이 아니라, 사용자에게 명시적으로 알려 줌으로써 기본적인 사용자 개인 정보 보호를 보장한다. 그리고 제안된 wear-UCAM은 센서와 서비스간의 자유로운 연결을 지원하기 위한 네트워킹 기능, 센서와 서비스간의 통신을 통합된 형태로 보장할 수 있는 정형화된 컨텍스트의 표현 방법을 포함하고 있다. 뿐만 아니라, 사용자의 생체 신호를 추출하고 분석할 수 있는 센서와 다양한 센서로부터 추출된 데이타를 분석하고 가공하여 컨텍스트화할 수 있는 다양한 컴포넌트들도 포함하고 있다. 따라서, 제안된 wear-UCAM은 유비쿼터스 컴퓨팅 환경에서 착용형 컴퓨팅에 필요한 사용자 중심의 컨텍스트기반 어플리케이션 모델이다. 본 논문에서는 제안된 wear-UCAM의 설계와 디자인된 wear-UCAM의 구체적인 구현 방법 및 실험에 대해서 자세히 설명한다. 또한, 제안된 wear-UCAM의 유용성을 검정하기 위해서 착용형 컴퓨터로 무선 네트워크가 가능한 PDA를 사용하여 wear-UCAM을 실험하였다.

본 논문의 구성은 다음과 같다. 2장에서는 제안된 착용형 컴퓨팅을 위한 어플리케이션 모델의 구조 및 구성요소들에 대해서 설명한다. 3장에서는 제안된 wear-UCAM의 실제 구현을 통한 실험 결과를 기술하고, 4장에서는 제안된 모델에 대한 결론과 향후 과제에 대해서 언급한다.

2. wear-UCAM의 구조 및 구성 요소

그림 1은 wear-UCAM의 구조를 개념적으로 표현한 그림이다.

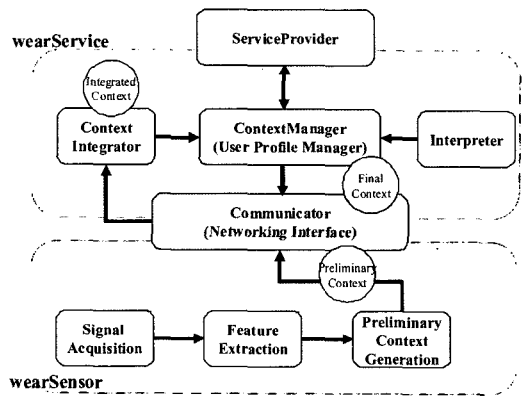

그림 1 제안된 wear-UCAM의 개념 구조도

그림 1과 같이, 제안된 wear-UCAM은 센서와 서비스를 분리하도록 하였을 뿐만 아니라 각각의 센서와 서비스가 네트워킹 기능을 포함하고 있다. 따라서 제안된 모델은 센서와 서비스가 독립적으로 동작하기 때문에 착용형 컴퓨팅 환경처럼 컴퓨팅 자원이 비교적 제한적인 환경에서도 동작하도록 보장 한다. 제안된 wear-UCAM은 ubi-UCAM[5,6]과 같이 정형화된 컨텍스트를 활용하여 센서와 서비스 간에 통신이 가능하게 하였다.

또한, 사용자 컨텍스트을 파악하기 위한 일련의 컴포넌트들은 ubi-UCAM과 동일하게 구성되어있다. 하지만, 사용자의 개인 정보와 관련된 컨텍스는 사용자 Profile (사용자 개인 정보의 보관, 관리 및 갱신이 가능한 객체 형태)을 활용하여 다른 착용형 컴퓨터나 환경으로부터 보호할 수 있는 부분은 wear-UCAM만의 특징이다[12].

2.1 wear-UCAM의 구조

wear-UCAM은 UCAM[1]의 일부분이며, 착용형 컴퓨팅 환경에서 필요한 다양한 컴포넌트를 패키징한 것이다. 그림 2는 UCAM의 전체 구조에서 wear-UCAM과 관련된 것들을 계층적 구조로 표현한 그림이다.

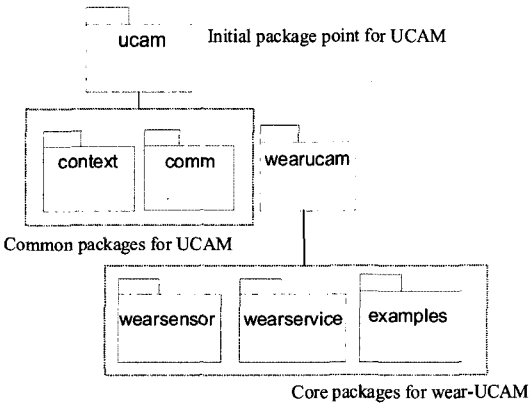

그림 2 wear-UCAM의 계층적 구조

그림 2에서 보듯이, wear-UCAM은 UCAM에서 사용하는 컨텍스트 정형화와 관련된 context패키지와 센서와 서비스에 네트워킹 기능을 제공하는 comm패키지를 공유한다. wear-UCAM의 중요 패키지는 wear-sensor, wearservice패키지로 wearucam으로 패키지화 되어 있다. 그림 2와 같이 wearucam패키지에는 착용형 컴퓨팅에서 활용 가능한 센서와 서비스를 충분히 지원하기 위해서 필요한 기능만을 추출하여 추상화(Abstract Class) 작업을 하였다. 예를 들면, wear-UCAM에서의 Sensor모듈은 센서로 부터 신호를 추출하고, 추출된 신호를 분석하여 초벌 컨텍스트(Preliminary Context)를 생성하는 역할을 담당한다. 따라서 wear-UCAM의 센서를 구현하기 위해서는 위의 3가지 기능을 반드시 구현하도록 개발자들에게 요구한다. 이와 관련된 자세한 내용은 다음 절의 처리 추상화(Procedural Abstraction)에서 자세히 설명한다.

결국, wear-UCAM의 구조는 착용형 컴퓨팅 환경에서 컨텍스트 기반 어플리케이션을 개발을 용이하게 할 뿐만 아니라, 정형화된 컨텍스트와 동일한 네트워크 기능을 통해서 유비쿼터스 컴퓨팅 환경의 어플리케이션과도 손쉽게 융합될 수 있는 구조이다.

2.2 컨텍스트 처리 과정의 추상화

제안된 wear-UCAM에서는 앞에서 설명한 것처럼 다양한 디자인 패턴을 활용하여 착용형 컴퓨팅을 위한 컨텍스트기반 어플리케이션 개발을 용이하도록 제안하였다. 하지만, 센서로부터 추출된 신호를 활용하여 컨텍스트를 생성하고 이용하기 위해서는 컨텍스트를 처리하는 일련의 과정이 필요하다. 따라서 제안된 wear-UCAM에서는 이와 같은 일련의 컨텍스트 처리 과정과 활용 과정을 처리 추상화(Procedural Abstraction)의 개념을 적용하여 제안 한다[14]. 그림 3은 컨텍스트 활용 과정의 추상화를 보여주는 예이다.

컨텍스트 처리과정에 있어서 추상화 기법은 Context Toolkit(CTK)에서도 활용된 기법이다[7]. CTK에서 활용된 추상화 기법은 센서로부터 추출된 컨텍스트가 응용 프로그램에서 사용되기 위해서 필요한 각 컴포넌트들을 개념적으로 추상화한 것이다. 따라서, 개발자들이 보다 손쉽게 컨텍스트를 활용할 수 있도록 하였다. 하지만, 제안된 wear-UCAM에서의 컨텍스트 처리 과정의 추상화는 그와 같은 기능뿐만 아니라 컨텍스트 활용 과정 추상화도 함께 제공한다.

그림 3은 wear-UCAM의 wearservice패키지에 구현된 Service와 ServiceProvider이다. wear-UCAM에서 서비스는 컨텍스트를 처리하는 모든 컴포넌트들을 포함하고 항상 서비스 대기 상태가 되도록 정의 한다. 그리고 특정 서비스(예를 들면, 사용자의 스트레스 정도에 따른 음악 연주)를 구체적으로 제공하는 부분을 ServiceProvider로 구현하도록 정의 한다. 따라서 이와 같은 서비스 실행의 요구 조건을 만족시키기 위해서 그림 3과 같이 설계하였다. 제안된 wear-UCAM의 Service와 ServiceProvider의 설계에서처럼 구체적인 구현 방법은 상이하지만, 컨텍스트를 활용하여 서비스를 제공하는 과정이 동일하기 때문에 처리 과정을 충분히 추상화시킬 수 있다. 처리 과정의 추상화가 갖는 장점은 동일한 처리 과정을 보장하면서도 다양한 구현이 가능한 점이다. 그리고 wear-UCAM에서는 추상화 개념을 확장하여 서비스를 다양하게 분류하였다[15]. 예를 들면, 사용자의 컨텍스트에 따라서 반응하는 TV서비스의 경우, 사운드뿐만 아니라 시각적 효과도 사용자에게 제공하기 때문에 사운드 서비스와 비쥬얼(Visual, Display) 서비스에 포함된다. 이는 특정 서비스가 하나의 서비스

1) UCAM은 Unified Context-aware Application Model로서, 본 연구실에서는 홈 환경 중심의 유비쿼터스 컴퓨팅환경을 목적으로 한 ubi-UCAM, 사용자 중심의 착용형 컴퓨팅 환경을 목적으로 한 wear-UCAM, 그리고 현실과 가상공간을 접목하려고 한 vr-UCAM이 각각 개발 중에 있다.

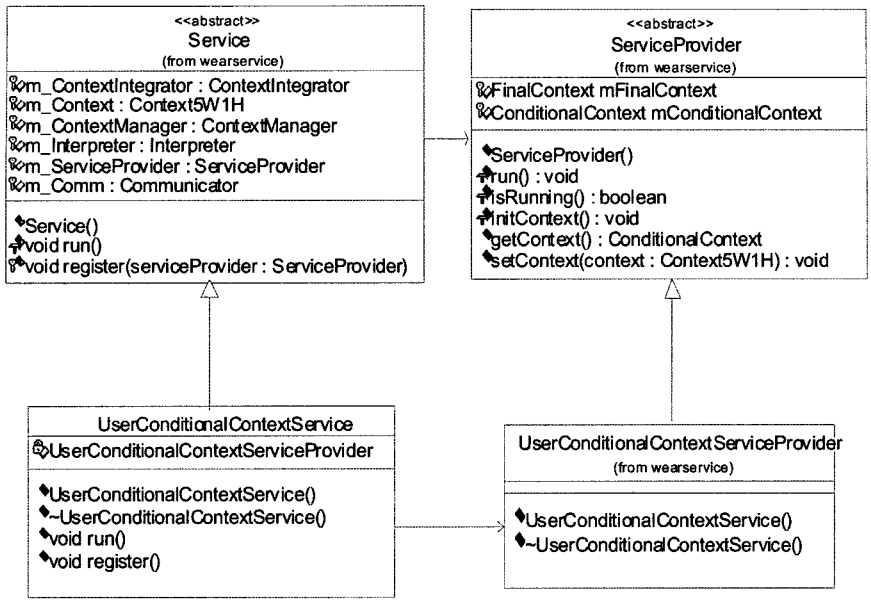

그림 3 컨텍스 활용 과정의 추상화 예

형태로 귀속되는 것이 아니라 다중적 의미를 갖는 서비스 형태가 가능하다. 따라서 추상화 기법은 wear-UCAM과 같이 어플리케이션 모델에는 필요한 기법 중 하나이다.

2.3 wear-UCAM의 컴포넌트들과 기능

제안된 wear-UCAM의 기본 컴포넌트들은 앞서 언급하였듯이 ubi-UCAM과 동일한 구성이다[5,6]. 뿐만 아니라, CTK의 각 컴포넌트들과도 유사 하다[7]. 예를 들면, CTK의 Widget은 센서로부터 입력된 신호를 Interpreter를 통해서 적절한 컨텍스트를 생성한다. CTK의 Aggregator는 다양한 Widget들을 통해 생성된 컨텍스트들이 컨텍스트 인식 어플리케이션에서 사용되기 위해서 적절한 통합과 관리를 한다. 하지만, 제안된 wear-UCAM에서는 CTK의 구성 요소들 보다 기능적인 요소들로 분리하여, 개발자들이나 사용자들에게 보다 쉽게 컨텍스트를 활용하고, 서비스를 제공받을 수 있게 한다.

그림 1에서처럼 wear-UCAM은 다양한 센서들로 부터 생성된 컨텍스트를 응용 프로그램에서 사용하기 위해서 일련의 컨텍스트 처리 과정이 필요하다. 표 1은 wear-UCAM에서 컨텍스트 처리 과정에 필요한 컴포넌트들을 센서와 서비스로 구분하여 나타낸다.

생체신호의 추출 및 분석은 사용자와 가장 밀접한 센싱 정보이며, 사용자 컨텍스트를 구성하는데 있어서 주요한 요소이다. 따라서 제안된 wear-UCAM에서는 센서 컴포넌트를 활용하여 구현된 wearBioSignalSensor (Sensor의 인스턴스)를 통해서 사용자의 생체신호를 추출하고 추출된 생체신호로부터 연산된 특징 값들을 정수 혹은 실수 형태로 변환한다. 이와 같이 변환된 특징 값들은 초벌 컨텍스트(Primitive Context)로 변환된다. 변환된 초벌 컨텍스트는 네트워크 단을 통해서 다른 센서나 서비스로 전송된다[13]. 센서나 서비스로 부터 추출된 컨텍스트들을 동적으로 전달하기 위해서 comm패키지에 있는 Communicator컴포넌트를 사용한다. Communicator에는 컨텍스트 필터링, 동적 멀티케스팅 그룹

표 1 wear-UCAM의 컴포넌트와 역할

구분	컴포넌트	역할
센서	*Sensor**	wearSensor의 기본 기능 제공 (신호처리, 특징추출, 초벌컨텍스트 생성)
네트워크	Communicator	센서와 서비스간 동적 연결
서비스	*Service**	wearService의 기본 기능 제공 (ServiceProvider등록, 서비스 분류)
	ContextIntegrator	초벌 컨텍스트분석을 통한 통합 컨텍스트생성
	ContextManager	통합 컨텍스트와 사용자의 선호도를 비교하여 최종 컨텍스트 생성
	Interpreter	사용자의 개인 정보와 관련된 컨텍스관리
	*ServiceProvider**	컨텍스트를 활용하여 착용형 컴퓨팅을 위한 서비스를 쉽게 구현할 수 있는 기능 제공

**는 추상화 클래스를 나타낸다.*

생성 및 컨텍스트 전달에 필요한 다양한 네트워크 기능들이 포함되어 있다[16]. 서비스는 센서로부터 생성된 초벌 컨텍스트를 컨텍스트 처리 및 활용 과정을 통해서 사용자에게 적합한 서비스를 제공한다. 이때, 서비스단의 ContextIntegrator는 센서나 서비스로부터 컨텍스트화된 사용자 정보, 즉, 사용자의 생체신호 정보 및 서비스 정보를 통합하여 통합 컨텍스트(Integrated Context)를 생성한다. 초벌 컨텍스트는 센서로부터 추출된 데이타와 약간의 처리과정을 거친 정보로서 사용자 중심의 서비스를 제공하기에는 불충분한 형태의 사용자 컨텍스트이다. 따라서 wear-UCAM의 컴포넌트인 Context-Integrator는 초벌 컨텍스트와 같이 기초적인 사용자의 컨텍스트 정보를 통합 알고리즘에 기반하여 통합 컨텍스트를 생성한다[17]. 따라서 통합 컨텍스트는 초벌 컨텍스트보다 사용자 중심의 컨텍스트로 표현되며, 이후 응용 서비스에서도 사용자의 의도를 반영할 수 있는 사용자 컨텍스트의 기본 단위가 된다. 그리고 제안된 wear-UCAM의 ContextManager에서는 사용자의 개인 정보의 인터페이스 역할을 하는 Interpreter로 부터 사용자의 개인 선호도와 ContextIntegrator로부터 받은 통합된 컨텍스트를 비교하여 최종 컨텍스트(Final Context)를 생성한다[18]. 최종 컨텍스트는 실제 응용 서비스가 동작하기 위한 기본 파라미터를 제공한다. wear- UCAM에서 Interpreter는 사용자에게 직접적인 입력을 요구하거나, 현재 서비스 가능한 서비스 리스트와 실행되는 서비스의 출력을 화면에 보여주는 역할을 한다. 그리고 ContextManager는 통합 컨텍스트를 활용하여 사용자의 개인 정보를 갱신하고 관리하는 역할을 한다.

본 논문에서는 wear-UCAM의 각 컴포넌트들이 ubi-UCAM에서 제안된 컨텍스트 정형화 기법을 이용하여 통신하도록 하였다. 하지만, 착용형 컴퓨팅 환경에 맞게 ubi-UCAM에서 제안된 컨텍스트 정형화 방법을 수정하였다. 예를 들면, 사용자 생체 신호와 관련된 항목들이 컨텍스트 정형화에 추가 되었다.

3. 실험 결과

현재 ubiHome에서 사용가능한 서비스들을 wear-UCAM이 탑재된 PDA기반의 사용자 인터페이스로 wear-UCAM을 실험하였다. 그림 4와 표 2는 본 논문에서 제안된 wear-UCAM을 구현하기 실험하기 위한 실험 환경과 관련된 정보를 나타낸다.

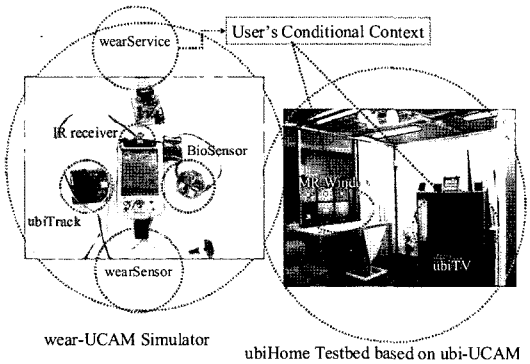

그림 4 wear-UCAM을 테스트한 PDA와 실험 환경

그림 4에서 처럼, 사용자의 스트레스 정도를 나타낼 수 있는 생체신호(BioSensor)와 사용자의 위치를 나타낼 수 있는 위치 센서(ubiTrack[19])를 wear-UCAM의 센서로 사용하였다. 그리고, 실험을 위해서 ubiHome에서 활용가능한 MR Window 서비스와 ubiTV 서비스를 활용하였다. 테이블 2는 wear-UCAM 시뮬레이터의 상세정보와 사용된 센서와 서비스의 하드웨어, 소프트웨어 정보를 나타낸다.

제안된 wear-UCAM의 툴로써의 효율성을 보이기위해서, 가상 센서와 서비스를 이용하였다. 하지만, 본 실험에서는 ubi-UCAM을 이용한 서비스를 제외하고, wear-UCAM과 관련된 부분만 실험하였다. 센서로부터 서비스에 이르는 컨텍스트 처리 과정의 시간은 다음과 같이 나타낼 수 있다.

$$FC(\Delta T) = PC(\Delta T_1) + Network(\Delta T_2) + IC(\Delta T_3) + UCC(\Delta T_4) \quad (1)$$

표 2 실험 환경 명세표

하드웨어		
장비	**기능 및 목적**	**비고**
iPAQ h5550	wear-UCAM Simulator	PocketPC 2003, RAM: 128MB, ROM: 48MB
BioSensor	Extract physiological signal	Bluetooth
ubiTrack	Track user's location	Serial communication
소프트웨어		
Java	Desktop development toolkit	J2SE (Java1.5)
Personal Java	PDA development toolkit	Java1.1.8 compatible
wear-UCAM	Toolkit for wearable computing	v0.9a1 (118KB .jar)

식 (1)에서 FC는 최종 컨텍스트를 추출하는데 소요된 시간을 나타내며, 이는 각 컴포넌트에서 소요된 시간의 합으로 표현된다. 각 항들은 초벌 컨텍스트 추출 과정(PC), 네트워크를 통한 전송 시간(Network), 통합 컨텍스트 추출 과정(IC), 그리고 사용자의 조건 컨텍스트 추출 과정(UCC)를 나타낸다.

하지만, wear-UCAM은 유비쿼터스 컴퓨팅 환경과 같이 분산 컴퓨팅 환경을 목적으로 디자인되었고, 각 컴포넌트들 자체가 쓰레드로 구현되어있기 때문에, 각 컴포넌트에서 소요되는 시간을 정확히 측정하는데 어려움이 따른다. 따라서, Java 모니터링 툴들인 jps, jstat, 와 visualgc를 사용하여 실험하였다[20]. 그림 5와 6은 각각 visualgc 모니터링 툴과 jstat 모니터링 툴을 나타낸다.

그림 5에서 보듯이, 서비스가 실행 중일 때 Java바이트 코드의 컴파일 시간이 실행 동안 추척된 형태로 시간이 표시된다. 이와 달리, 그림 6에서는 각 단계별로 컴파일 시간이 표시된다. visualgc 모니터링 툴과 달리, jstat 모니터링 툴은 많은 옵션을 제공하지만, 텍스트 기반의 모니터링 툴이다. 그림 6은 jstat 모니터링 툴을 나타낸다.

그림 5와 6은 컴파일 시간, 클래스 로딩 시간, 메모리 사용량 및 컨텍스트 추출에 소요된 시간에 관한 wear-UCAM 시뮬레이션 결과의 일부분을 보여준다. jstat 모니터링 툴을 통해서, 센서와 서비스를 실행 했을 때 다음과 같은 몇 가지 사실을 관찰 할 수 있었다. 우선, 센서와 서비스에서 실행시 컴파일 시간은 수행해야 할 컴파일 시간에 비례한다. 그림 5와 6에서 보듯이, 실행중인 서비스의 컴파일 수행 횟수가 267일 때, 컴파일 시간은 0.18ms이다. 하지만, 일정 시간이 경과하게 되면, 실행 중인 서비스나 센서의 컴파일 시간과 컴파일 수행 횟수가 수렴되는 것을 알 수 있다. 클래스 로더 시간의 경우에 있어서, 센서는 51개의 클래스가 실행 시 로딩되고, 로딩 시간은 0.01ms였다. 한편, 서비스는 95개의 클래스가 로딩되며, 로딩 시간은 0.02ms였다. 이는 wear-UCAM을 이용한 센서와 서비스의 경우, 시스템적인 측면에서 즉각적인 서비스가 가능함을 알 수 있다. 두 경우에 있어서, 로딩된 메모리는 각각 45.9KB와 91.7KB였다. 메모리 사용량과 관련해서, 응용 프로그램이 필요한 지속적인 메모리 영역을 측정하였다. 지속적인 메모리 영역은 8192KB로 고정적이며, 약 1MB정도의 메모리를 요구함으로 제한적인 컴퓨팅 환경에 적합함을 알 수 있다.

다음은 제안된 wear-UCAM의 유용성을 실험하기 위해서 센서로부터 센싱된 정보가 어떻게 컨텍스트로 도출되는지를 보여준다. wear-UCAM에서의 각 컴포넌트들 간에 사용되는 모든 데이타들은 초벌 컨텍스트, 통합

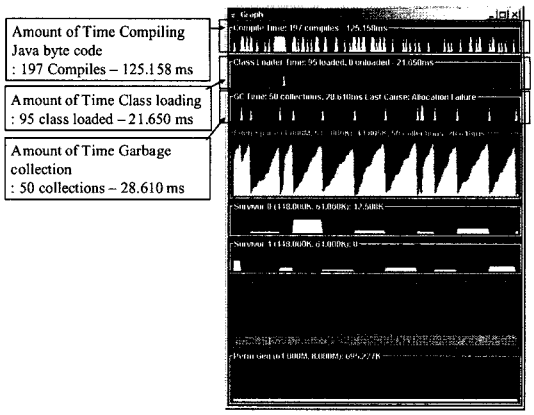

그림 5 visualgc 모니터링 툴(서비스 실행 예)

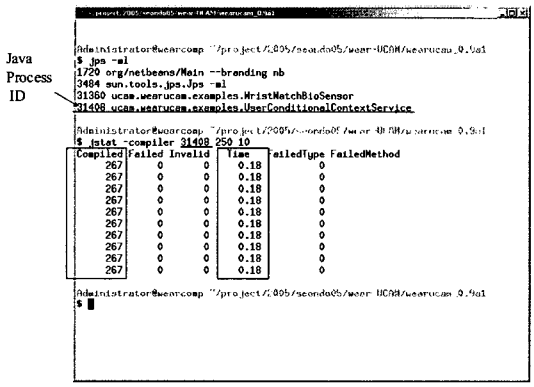

그림 6 jstat 모니터링 툴(서비스 실행 예)

컨텍스트, 그리고 최종 컨텍스트의 형태의 5W1H로 표현 된다[5,6]. 각 컨텍스트 표현 형태는 단순한 형태가 아닌 구조화된 형태로 표현된다. 표 3은 본 실험을 위해서는 2개의 센서를 사용하였고, 각 단계에서의 사용자 컨텍스트이다.

표 3에서처럼, wear-UCAM의 각 컴포넌트를 통해서 점진적으로 사용자나 개발자가 알기 쉬운 형태의 컨텍스트 표현됨을 알 수 있다. 표 3에서는 구조화된 컨텍스트의 일부만을 표현하였다.

본 실험을 통해서, wear-UCAM에서는 최종 컨텍스트 추출이 사용자의 조건 컨텍스트와 매우 밀접한 관계가 있음을 알 수 있었다. 이는 사용자의 조건 컨텍스트와 다양한 센서를 통해서 통합된 통합 컨텍스트가 반드시 일치해야만 최종 컨텍스트가 생성되기 때문이다. 따라서, ContextManager의 컨텍스트 매칭 알고리즘뿐만 아니라 ServiceProvider의 사용자 조건 컨텍스트 제공을 위해 사용자에게는 편리하고, 개발자들에게는 매칭 알고리즘을 유연하게 변경가능하도록 제공해야 한다. 또

표 3 wear-UCAM 컴포넌트에서의 컨텍스트 표현 예

구분	컴포넌트		컨텍스트 표현	비고
센서	BioSensor	Who	Name=Dongpyo, Priority=4	초벌컨텍스트
		What	SenosrID=1, SensorType=BioSensor, Accuracy=0x03	
		How	GSR.mean=14.37 GSR.var=3.0 PPG.HR_mean=63.0 PPG.HR_var=3.4 PPG.HR_LF_power=70.0, PPG_HRV_HF_Power=62.0 SKT=36.5	
	ubiTrack	Who	Name=Dongpyo, Priority=4	
		Where	IndoorLocation.x=140 IndoorLocation.y=160 SymbolicLocation=ubiHome Direction=0.0	
		What	SenosrID=1, SensorType=BioSensor, Accuracy=0x03	
서비스	ContextIntegrator	Who	Name=Dongpyo, Priority=4	통합컨텍스트
		When	130527(hh:mm:ss)	
		Where	IndoorLocation.x=140 IndoorLocation.y=160 SymbolicLocation=ubiHome Direction=0.0	
		What		
		How	PPG.HR_mean=63.0 PPG.HR_var=3.4 PPG.HR_LF_power=70.0, PPG_HRV_HF_Power=62.0	
		Why	Stress=1	
	ContextManager	Who	Name=Dongpyo, Priority=4	최종컨텍스트
		When	130527(hh:mm:ss)	
		Where	IndoorLocation.x=140 IndoorLocation.y=160 SymbolicLocation=ubiHome Direction=0.0	
		What	Service=MRWindow Service	
		How	PPG.HR_mean=63.0 PPG.HR_var=3.4 PPG.HR_LF_power=70.0, PPG_HRV_HF_Power=62.0	
		Why	Stress=1	

**GSR(skin conductance: 피부전도도), PPG(photoplethysmogram: 심전도), SKT(skin temperature: 체온)*

한, ContextIntegrator의 통합 방법도 센서로부터 추출된 초벌 컨텍스트의 내용이 부족하더라도 충분히 다른 부분들을 유추할 수 있는 형태로 제공될 필요가 있다. 이는 보다 정확한 최종 컨텍스트의 추출을 증가시키는 wear-UCAM의 핵심이다.

끝으로, 표 2에서 처럼, wear-UCAM 패키지 포팅에 사용된 환경은 WinCE 4.0(PocketPC 2003) 플랫폼과 Personal Java이다. wear-UCAM은 소스 코드 배포의 용이성과 코드의 구조화를 위해서 .jar 파일 형태로 구현되었다. 현재 구현된 코드의 패키지는 약 122 KB이다. 따라서, 착용형 컴퓨팅 환경과 같이 컴퓨팅 자원의 제약이 심한 환경에서 도 충분히 동작 할 만큼 용량은 적다. 하지만, 개발 중인 wear-UCAM에서는 비교적 용량이 큰 하드웨어 디바이스 드라이브(예를 들면, 생체신호 계측용 디바이스 드라이브)를 포함하지 않고 있기 때문이다. 향후 wear-UCAM의 개발에 있어서 패키지의 용량 문제는 동적 링크나 로딩 기법을 활용할 것이다. wear-UCAM의 구현 언어를 Java로 채택한 것은 다양한 플랫폼으로의 이식성이나 객체지향적인 개념을 표현하기에 적절한 언어이기 때문이다. wear-UCAM

개발을 Java언어로 시작하였으나, 개발 중 다소 어려움 점들이 몇 가지 있었다. PDA나 모바일상에서 동작하는 Java언어와 데스크탑 환경에서 동작하는 Java언어가 갖는 차이점 때문에, wear-UCAM의 여러 컴포넌트들은 제약 조건을 갖고 구현되었다. 향후 wear-UCAM을 보다 더 제한적인 착용형 컴퓨팅 환경에서 개발한다면, 심각한 문제로 대두 될 수 있다. 또 다른 문제는 센서로부터 실시간으로 신호를 처리해야 할 때, 가상 머신(Virtual Machine)에 기반한 Java는 그 효율성에서 C/C++와 비교해서 떨어진다. 하지만, 코드 튜닝과 같은 방법을 통해서 어느 정도는 효율성을 보장할 수도 있다. 향후 wear-UCAM의 개발은 J2ME에 기반하여 PDA뿐만 아니라 다양한 모바일 기기에서 실험할 계획이다. 하지만, wear-UCAM과 같이 착용형 컴퓨팅 환경에서 응용 프로그램을 개발하고자 할 때에는 반드시 앞서 지적된 부분들이 우선 고려되어야 한다.

4. 결론 및 추후 과제

본 논문에서는 착용형 컴퓨팅을 위한 정형화된 컨텍스트 인식 응용 모형인 wear-UCAM을 제안하였다. 제안된 wear-UCAM은 유비쿼터스 컴퓨팅 환경에서 사용자와 관련된 개인 정보를 센서로부터 획득하고 획득된 정보를 처리 및 분석해서 사용자의 컨텍스트에 따른 서비스를 제공할 뿐만 아니라, 개인 정보 보호의 기능도 가능한 컨텍스트 기반 어플리케이션 모델이다. 본 논문에서는 제안된 wear-UCAM에서 정의한 센서와 서비스 간에 필요한 컨텍스트 처리 과정의 추상화뿐만 아니라 wearSensor와 wearService로 센서와 서비스를 분리함으로써 다양한 센서와 응용 프로그램들 간의 독립성을 보장하였다. 하지만, 실제 착용형 컴퓨터에서 구현 및 테스트, 그리고 구체적인 컨텍스트 기반 어플리케이션을 실험하지 못하였다. 따라서 향후 제안된 모델을 소프트웨어 공학적인 측면에서의 분석뿐만 아니라 컴퓨팅 자원이 비교적 제한적인 착용형 컴퓨터에서의 실험과 분석이 요구된다.

참 고 문 헌

[1] M. Weiser, "Some computer science issues in ubiquitous computing," Communications of the ACM, vol. 36, no. 7, pp. 75-84, July 1993.

[2] Guanling Chen, and David Kotz. "A Survey of Context-Aware Mobile Computing Research," Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November, 2000.

[3] Bradley J. Rhodes, Nelson Minar and Josh Weaver, "Wearable Computing Meets Ubiquitous Computing: Reaping the best of both worlds," ISWC '99, San Francisco, CA, October 18-19 1999, pp. 141-149.

[4] Jennica Falk,Staffan Björk, "Privacy and information integrity in wearable computing and ubiquitous computing," Conference on Human Factors in Computing Systems archive CHI '00, pp.177-178, 2000.

[5] S.Jang, W.Woo, "ubi-UCAM: A Unified Context-Aware Application Model," Lecture Note Artificial Intelligence, Vol.2680, pp. 178-189, 2003.

[6] Y.Oh, C.Shin, S.Jang, W.Woo, "ubi-UCAM 2.0: A Unified Context-aware Application Model for Ubiquitous Computing Environments," ubiCNS, Proceeding CD, 2005.

[7] Dey, A.K., Salber, D. Abowd, G.D., "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," anchor article of a special issue on Context-Aware Computing Human-Computer Interaction (HCI) Journal, Vol. 16 (2-4), 2001, pp. 97-166.

[8] M. C. Mozer, "The Neural Network house: An environment that adapts to its inhabitants," In M. Coen (Ed.), Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, Menlo, Park, CA: AAAI Press., pp. 110-114, 1998.

[9] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. "Advanced interaction in context," In H.W. Gellersen, editor, Proc. of First International Symposium on Handheld and Ubiquitous Computing (HUC99), volume 1707 of LNCS, pages 89-101. Springer-Verlag, 1999.

[10] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software," Addison Wesley. October 1994.

[11] Fowler, Martin, "UML distilled: applying the standard object modeling language," Addison-Wesley, 2004.

[12] Y.Suh, W.Woo, "User Profile Management for Context-aware Applications for ubiHome environment," ubiCNS05, Proceeding CD, 2005.

[13] A.Choi and W.Woo, "Feature extraction for emotion analysis based on physiological signal pattern," KHCI2005, pp. 624-629, 2005.

[14] Wizard Book n. Hal Abelson's, Jerry Sussman's and Julie Sussman's Structure and Interpretation of Computer Programs(MIT Press, 1984; ISBN 0-262-01077-1).

[15] C.Shin and W.Woo, "Conflict Resolution among Users for Context-aware Media Services," KHCI2005, pp. 594-599, 2005.

[16] Sj.Oh, Y.Lee, and W.Woo, "Dynamic Network Reconfiguration for Seamless Interactions between Real and Virtual Environments," KHCI2005, pp.721-

726, 2005.
[17] Y.Oh, W.Woo, "A unified Application Service Model for ubiHome by Exploiting Intelligent Context-Awareness," UCS2004, pp.117-122, 2004.
[18] C.Shin, W.Woo, "Conflict Resolution Method using Context History for Context-aware Applications," Pervasive 2005 workshop, accepted, 2005.
[19] W.Jung, W.Woo, "Indoor orientation tracking using ubiTrack," ubiCNS2005, Proceeding CD, 2005.
[20] http://java.sun.com/performance/jvmstat/

홍 동 표
2001년 동아대학교 컴퓨터공학과 학사 2004년 광주과학기술원 정보통신공학과 석사. 2004년~현재 광주과학기술원 정보통신공학과 박사과정. 관심분야는 HCI, Information Retrieval, Context-awareness, Vision based User Interface, Wearable computing, Ubiquitous computing, VR, and Entertainment Computing 등

우 운 택
1989년 경북대학교 전자공학과 학사. 1991년 포항공과대학교 전기전자공학과 석사 1998년 University of Southern California, Electrical Engineering-System 박사. 1991년~1992년 삼성종합기술연구소 연구원. 1999년~2001년 ATR MIC Labs. 객원 연구원. 2001년~현재 광주과학기술원 정보통신공학과 조교수. 관심분야는 3D computer vision and its applications including attentive AR and mediated reality, HCI, affective sensing and context-aware for ubiquitous computing 등