

유비쿼터스 컴퓨팅 환경을 위한 컨텍스트 기반 애플리케이션 구조

장세이, 우운택
광주과학기술원 정보통신공학과 U-VR 연구실
{jangsei, wwoo}@kjist.ac.kr

Architecture of Context based Application in Ubiquitous Computing Environment

Seiie Jang, Wontack Woo
KJIST U-VR Lab.

요약

유비쿼터스 컴퓨팅이 제공하는 기반구조를 바탕으로 사용자 및 사용자 주변에 대한 정보를 생성하고, 이 정보를 기반으로 사용자가 요구하는 서비스를 제공하는 것은 편리성을 증가시킬 수 있다. 본 논문에서는 유비쿼터스 컴퓨팅 환경에서의 컨텍스트 기반 애플리케이션 구조를 제안한다. 제안된 구조는 SWIH 형태의 컨텍스트를 사용하여 센서와 애플리케이션의 상호 의존성을 최소화하며, 동시에 애플리케이션은 필요한 컨텍스트를 SWIH에서 직접 생성하여 애플리케이션 상태에 따라 최적화된 컨텍스트를 사용할 수 있다. 제안된 구조는 본 연구실이 지능형 가정 환경을 실험하기 위해 구축한 ubiHome 테스트 베드의 센서와 애플리케이션을 구현하는데 적용되었으며 더 나아가 유비쿼터스 컴퓨팅 환경이 갖추어진 가정, 사무실 등의 일상 생활 공간에서 컨텍스트 기반 애플리케이션을 구축하기 위한 구조로 사용될 수 있다.

Keyword : Context based Service, Ubiquitous Computing, Smart Environment

1. 서론

유비쿼터스 컴퓨팅[1][2]은 사용자가 언제 어디서나 일상 생활 속에 편재해 있는 컴퓨팅 자원을 이용하여 다양한 서비스를 제공하는 기반구조이다. 컨텍스트 기반 애플리케이션은 유비쿼터스 컴퓨팅의 기반구조를 이용하여 사용자 및 사용자 주변에 대한 정보 즉, 컨텍스트를 생성하고 그 컨텍스트를 기반으로 사용자에게 필요한 정보나 서비스를 제공한다. 따라서 유비쿼터스 컴퓨팅 환경에서의 컨텍스트 기반 애플리케이션 구조는 컨텍스트에 대한 정보를 제공하는 부분(센서)과 컨텍스트를 이용하여 서비스를 제공하는 부분(애플리케이션)이 개별적 구성 요소로써 존재한다. 그러나 센서

와 애플리케이션은 컨텍스트를 필요로 할 때 상호 작용을 하는 분산 개체적 특징을 갖는다. 이를 위해 센서와 애플리케이션은 컨텍스트를 생성하고 관리하는 중간 매개체를 사용하지 않고, 직접적인 상호작용을 하는 메커니즘을 요구한다.

한편 컨텍스트 기반 애플리케이션에 대한 연구는 자동화(automation), HCI 등의 다양한 분야에서 이미 진행되고 있다. 예를 들어 Colorado Univ.의 ACHE(Adaptive Control of Home Environment)는 신경망을 이용하여 사용자의 반복되는 행동 패턴을 컨텍스트로 파악하여 실내 온도, 조명, 샤워 물 온도 등의 거주 환경을 조정한다[3]. 또한 Rutgers Univ.의 MIM(Multimedia Interface Manager)는 사용

* 본 논문은 정보통신부의 기초기술지원사업과 교육 인적 자원부의 BK21 사업에 의해 수행됨

자의 멀티모달 (multimodal : 시각, 촉각, 청각)을 컨텍스트로 이용하여 사용자와 컴퓨터간의 의사 전달 환경을 제공한다[4].

그러나 이러한 컨텍스트 기반 애플리케이션들이 센싱, 프로세싱, 네트워킹 등의 기술들을 보다 효과적으로 사용할 수 있는 환경을 필요로 함에 따라 점차 유비쿼터스 컴퓨팅이 제공하는 기반 구조를 이용하는 연구로 집중되고 있다. 본 논문은 이러한 흐름에 맞춰 유비쿼터스 컴퓨팅 환경의 컨텍스트 기반 애플리케이션 구조를 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 유비쿼터스 컴퓨팅 환경을 위한 컨텍스트 기반 애플리케이션 구조가 해결해야 할 문제점에 대해 살펴본다. 그리고 3 장에서는 이러한 문제점을 해결한 컨텍스트 기반 애플리케이션 구조를 제안하고, 4 장에서는 제안된 구조의 구현 상황에 대해 설명한다. 마지막으로 5 장에서는 앞으로의 연구 방향에 대해 살펴본다.

2. 컨텍스트 기반 애플리케이션의 문제점

유비쿼터스 컴퓨팅 환경을 위한 컨텍스트 기반 애플리케이션 구조를 디자인할 때 고려해야 할 점들은 다음과 같다.

• 모든 서비스가 공통으로 사용 가능한 컨텍스트를 생성할 수 있는가?

모든 애플리케이션들이 컨텍스트를 공통으로 사용하기 위해서는 먼저 컨텍스트에 대한 정의가 필요하다. Schilt 등은 이미 컨텍스트를 사용자와 오브젝트에 관련된 신원 및 오브젝트 정보로 정의하였다 [5]. 또한 Dey 등은 컨텍스트를 사용자, 공간, 오브젝트 등의 개체와 관련된 모든 정보라고 정의하였다[6]. 그러나 이와 같은 정의는 애플리케이션이 요구하는 컨텍스트 표현 방법에 대한 일관성을 보장하지 못한다. 즉, 애플리케이션에 따라 같은 종류의 컨텍스트라도 표현 방법이 다르기 때문에 애플리케이션이 공통으로 사용할 수 있는 컨텍스트를 표현하는 것은 쉬운 것이 아니다. 그러나 컨텍스트를 SWIH: who, what, where, when, why, how 형태로 나타내면 사용자 및 사용자 주변에 대한

정보를 포괄적으로 제공할 수 있으며, 또한 애플리케이션은 SWIH로부터 요구하는 컨텍스트를 직접 생성할 수 있다. 이러한 관점에서 Jang 등은 컨텍스트를 SWIH로 나타낼 수 있다고 가정하고 애플리케이션에 따라 SWIH의 조합으로 구성된다고 정의하였다[1][2]. 따라서 센서는 규격화된 SWIH를 제공함으로써 컨텍스트 표현의 일관성을 보장하고, 애플리케이션은 SWIH로부터 필요한 컨텍스트를 생성함으로써 모든 서비스가 공유할 수 있는 컨텍스트의 생성을 보장할 수 있다.

• 컨텍스트 생성을 위한 센서와 컨텍스트를 사용하는 애플리케이션 사이에 독립성을 제공해야 하는가?

센서가 애플리케이션에 종속되면 애플리케이션이 요구하는 형태로 컨텍스트를 즉시 제공할 수 있지만 애플리케이션이 변경되면 센서에 대한 처리부분 역시 수정되는 문제점이 있다. 또한 센서가 애플리케이션에 독립되면 애플리케이션의 변경이 쉽게 이뤄질 수 있지만 애플리케이션에 맞춰진 컨텍스트를 생성하는 것이 어렵다. 따라서 애플리케이션에 맞는 컨텍스트를 생성하면서 센서와 애플리케이션의 종속관계가 최소로 유지되는 것이 바람직하다. 이를 위해 센서는 컨텍스트 생성에 필요한 기본적인 정보를 규격화하여 제공함으로써 임의의 애플리케이션과 연결이 가능하다. 또한 애플리케이션은 이러한 정보로부터 컨텍스트를 직접 생성함으로써 애플리케이션에 최적화된 컨텍스트를 사용할 수 있다.

3. 컨텍스트 기반 애플리케이션 모델

제안된 컨텍스트 기반 애플리케이션 모델은 애플리케이션들이 공유할 수 있는 SWIH 형태의 컨텍스트를 제공하고, 동시에 애플리케이션과 센서가 독립된 구조를 나타낸다.

제안된 모델은 그림 1 과 같이 유비서비스 (ubiService)와 유비센서 (ubiSensor)로 구성된다. 각 구성요소는 네트워킹 및 프로세싱 모듈을 포함하여 다른 구성 요소와 서로 연결된다. 유비센서는 모든 유비서비스가 사용될 수 있는 규격화된 초기 컨텍스트 (primitive SWIH)를 생성함으로써 임의의

유비서비스와의 연결이 가능하다. 또한 유비서비스는 여러 개의 초기 컨텍스트로부터 서비스를 실행시키기 위한 최종 컨텍스트(final SWIH)를 결정함으로써 유비서비스가 요구하는 컨텍스트를 직접 생성한다. 그러므로 유비센서는 SWIH 형태의 컨텍스트를 사용하여 사용자와 사용자 주변에 대한 포괄적 정보를 초기 컨텍스트로 제공함으로써 유비서비스와의 임의의 연결성을 보장하고, 유비서비스는 초기 컨텍스트와 유비서비스의 상태 정보로부터 최종 컨텍스트를 생성함으로써 최적화된 컨텍스트의 사용을 제공한다.

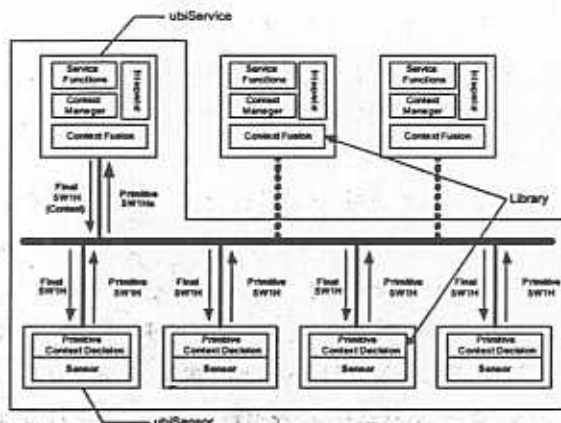


그림 1. 컨텍스트 기반 애플리케이션 모델

3-1. 유비센서

유비센서는 그림 1 과 같이 사용자 및 사용자 환경에 대한 변화를 감지하는 센서모듈과 감지된 정보를 초기 컨텍스트로 생성하는 초기 컨텍스트 결정 (primitive context decision) 모듈 부분으로 구성된다. 초기 컨텍스트 결정 부분은 센서모듈에서 전달되는 정보를 분석하여 라이브러리에 있는 SWIH 와 가장 가까운 값을 찾아낸다. 이때 사용되는 센서모듈에 따라 SWIH 의 일부 또는 전체가 결정되며, 또한 유비서비스로부터 전달된 최종 컨텍스트를 참조하여 보다 정확한 결정한다. 그리고 유비센서는 이러한 결정을 할 때 마다 푸싱 (pushing) 기술을 이용하여 그림 2 와 같은 초기 컨텍스트를 애플리케이션으로 전달한다. 그러나 하나의 유비센서가 초기 컨텍스트의 SWIH 를 모두 결정할 필요는 없다. 즉 센서모듈의 특징에 따라 SWIH 중 결정 가능한 부분을 생성하고, 나머

지 부분은 '-'로 처리한다.

Primitive SWIH	who+what+where+when+how+why+n
Final SWIH	Who, What, Where, When, Why, How : are enumerated in library

그림 2. 컨텍스트 메시지 형식

3-2. 유비서비스

유비서비스는 컨텍스트 통합 (context fusion), 컨텍스트 관리 (context manager), 해석기 (interpreter), 그리고 서비스 (service function)등의 모듈로 구성된다.

컨텍스트 통합 모듈은 유비서비스와 연결된 여러 개의 유비센서로부터 일정한 시간 간격으로 초기 컨텍스트를 수집한다. 그리고 초기 컨텍스트의 각 항목에 대한 의사 결정 방법을 적용하여 최종 컨텍스트를 생성한다. 현재 구현된 의사 결정 방법으로는 Who 와 What 을 위해서는 Voting, Where 와 When 을 위해서는 Mean, How 를 위해서는 OR, 그리고 Why 를 위해서는 Neural Net 등을 사용한다. 특히 컨텍스트 통합 모듈은 각각의 유비센서가 부분적으로 생성한 초기 컨텍스트를 통합하여 SWIH 가 모두 갖춰진 형태의 최종 컨텍스트를 생성한다. 이것은 유비서비스와 연결된 유비센서로 멀티캐스팅 (multicasting) 된다.

컨텍스트 관리 모듈은 현재의 최종 컨텍스트와 일치 되는 컨텍스트 조건 (condition)을 검색한다. 컨텍스트 조건은 특정 서비스 모듈과 연결된다. 따라서 최종 컨텍스트와 일치된 컨텍스트 조건이 검색되면 해당되는 서비스 모듈이 자동으로 실행된다. 컨텍스트 조건은 그림 3 과 같이 해쉬 테이블 (Hash table)을 이용하여 구성되어 있다. 해쉬 테이블은 컨텍스트 조건을 키 (key)로, 그리고 서비스 정보를 값 (value)으로 사용한다. 특히 해쉬 테이블은 보다 빠른 시간으로 컨텍스트 조건의 검색을 보장한다.

Hash Table	
Condition #1	Information for Function #1
Condition #2	Information for Function #2
...	...
Condition #n	Information for Function #n

그림 3. 컨텍스트 조건 검색을 위한 해쉬 테이블

해석기는 서비스와 컨텍스트 조건을 연결하여 해석 테이블을 생성하는 역할을 담당한다. 이러한 컨텍스트 조건은 사용자에게 의해 설정될 수 있으며, 특히 사용자 프로파일 정보를 활용하여 자동으로 설정되는 기능을 제공한다.

3-3. 네트워킹

유비센서는 룩업서비스(lookup service)를 제공하는 네트워크에 연결된다. 따라서 유비서비스는 필요한 유비센서에 대한 정보를 룩업서비스로 문의하고, 룩업서비스는 사용 가능한 유비센서에 대한 정보를 유비서비스로 전달한다. 유비서비스는 이 정보를 이용하여 유비센서와 직접 연결하여 메시지를 교환한다. 그리고 유비서비스가 종료되면 유비센서와의 연결을 제거한다. 이때 유비센서는 연결상태에 대한 정보를 룩업서비스에 즉각 보고한다.

유비서비스와 유비센서의 메시지 교환 메커니즘은 푸싱과 멀티캐스팅이다. 푸싱은 유비센서가 초기 컨텍스트를 생성하면 유비서비스로 즉시 전달하고, 멀티캐스팅은 유비서비스와 연결된 유비센서로 최종 컨텍스트를 전달한다.

4. 구현 및 토의

본 연구실에서는 스마트 가정 환경을 실험할 수 있는 'ubiHome' 이라는 유비쿼터스 컴퓨팅 테스트베드를 구축하고 있다. ubiHome 은 컨텍스트를 생성하기 위한 다양한 종류의 유비센서들과 유비서비스로 구성된다. ubiHome 에서 제공되는 서비스는 사용자 신원 정보, 사용자의 자세, 사용자의 제스처, 사용자의 위치 등의 컨텍스트를 사용하여 영화를 상영하는 Movie Player가 구현되었다.

Movie Player 는 그림 4 와 같이 ubiKey(사용자 신원정보), ubiFloor(사용자 위치 정보), spacesensor(사용자 제스처), coach sensor(사용자 자세) 등의 유비센서들과 연결되어 있다. 특히 사용자 제스처, 사용자 자세 등의 정보는 유비센서와 유비서비스에서 5WIH 형태의 메시지를 관리하는 부분에 미리 정의 되어있다. 그러한 정보들은 사용자 이름, 위치 정보, 시간 정보가 일관성 있는 형태로 표현이 가능하지만 사용자 제스처, 자세 등의 표현은 일관성 있는 형태로 나타내기 힘들기 때문이다. 따라서 사용자 제스처와 자세 정보를 라이브러리에 포함시켰으며, 유비센서와 유비서비스들은 라이브러리를 업데이트함으로써 보다 다양한 정보를 사용할 수 있다.

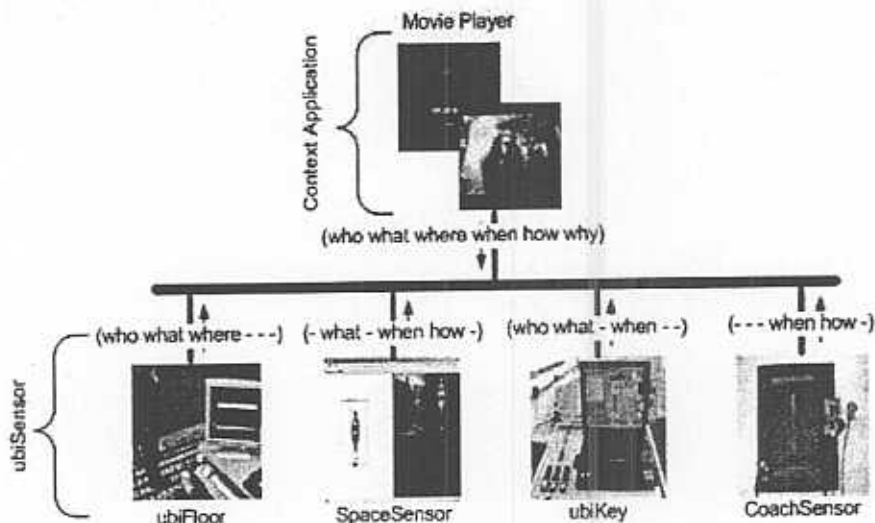


그림 4. 컨텍스트 기반 애플리케이션의 구현 예(Movie Player)

이러한 컨텍스트를 이용하여 다음과 같은 서비스를 제공한다. 사용자 신원정보에 따라 상영될 영화 목록이 다르게 나타나며 사용자가 스크린과의 위치에 따라 소리가 조정되며 사용자가 쇼파에 앉거나 누울 때에 따라 화면이 반전되어 나타나며, 영화 상영을 위한 제어를 사용자의 제스처로 하는 것 등이 있다.

Movie Player 는 그림 5 와 같이 Start, Selection, Play, Volume, Pause, FF, RW 등의 내부 상태를 갖는다. 그리고 동일한 최종 컨텍스트라도 Movie Player 의 현재 상태에 따라 서로 다른 서비스를 제공한다. 즉 Movie Player 의 상태에 따라 최종 컨텍스트가 실행시키는 서비스가 실행됨으로써 보다 다양하게 최종 컨텍스트를 이용할 수 있다.

현재 구현된 Movie Player 는 그림 6 과 같은 구조로 구현되었다. 특히 컨텍스트 통합, 컨텍스트 관리, 해석기 등은 패키지로 구현되어 제공된다. 따라서 Movie Player 에 제공되는 서비스와 컨텍스트 조건을 위한 해쉬 테이블을 생성하기 위해 해석기를 변경하면 다른 서비스를 제공하는 유비서비스가 된다. 현재 Movie Player 와 비슷한 서비스를 제공하는 비디오, 오디오, TV 등의 가전 기기를 유비서비스로 제어하는 연구를 병행하고 있다.

5. 결 론

본 논문이 제안한 컨텍스트 기반 애플리케이션 구조는 유비센서와 유비서비스에 라이브러리 형태로 제공되는 규격화된 5W1H 형태의 초기 컨텍스트를 통해 여러 유비서비스들에 의해 공유되며 또한 각 서비스의 상태에 알맞은 최종 컨텍스트를 생성할 수 있는 구조를 제안하였다. 또한 기존의 컨텍스트 기반 애플리케이션 구조에서와 같이 컨텍스트가 센서에서 애플리케이션으로 일방적으로 전달하는 구조에서 벗어나, 애플리케이션에서 센서로 컨텍스트를 전달함으로써 보다 정확성 높은 컨텍스트를

를 결정하도록 하는 구조를 제공한다.

앞으로의 연구는 보다 다양한 애플리케이션의 개발을 통해 보다 많은 애플리케이션들에 의해 공유될 수 있는 규격화된 5W1H 의 표현방법을 체계적으로 연구하는 것과 함께 애플리케이션과 센서의 연결이 록업서비스를 이용하지 않는 매커니즘을 추가하는 것 등이 있다. 또한 애플리케이션과 센서에 제공되는 컨텍스트 생성 모듈을 보다 최소화하는 것이 진행되어야 한다.

6. 참고문헌

- [1] S. Jang, S. Lee, W. Woo, "Research Activities on Smart Environment", IEEK, Magazine, vol. 28, pp. 85-97, Dec. 2001
- [2] J. Yoon, S. Lee, Y. Suh, J. Ryu, W. Woo, "Information Integration System for User Recognition and Location Awareness in Smart Environment", KHCI 2002, Feb. 2002.
- [3] Mozer, M. C. "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.
- [4] I. Marsic, A. Medl, J. Flanagan, J. "Natural communication with information systems," Proceedings of the IEEE, vol.88, no. 8, pp. 1354 - 1366, Aug. 2000.
- [5] Schilit, B., Theimer, "M. Disseminating Active Map Information to Mobile Hosts", IEEE Networks, vol.8, no.5, pp. 22-32, 1994
- [6] Anind K. Dey, "Understanding and Using Context," Personal and Ubiquitous Computing, Special issue on Situated Interaction and Ubiquitous Computing, vol.5, no.1, 2001

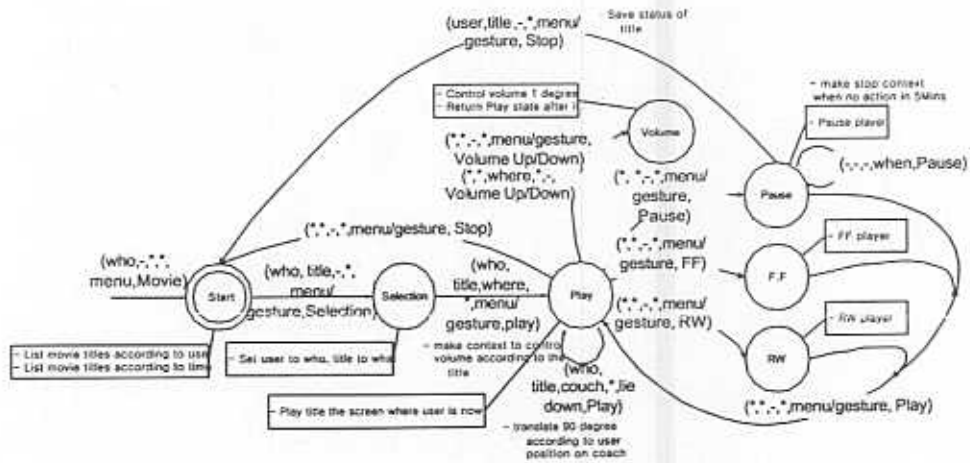


그림 5. Movie Player의 상태 및 최종 컨텍스트

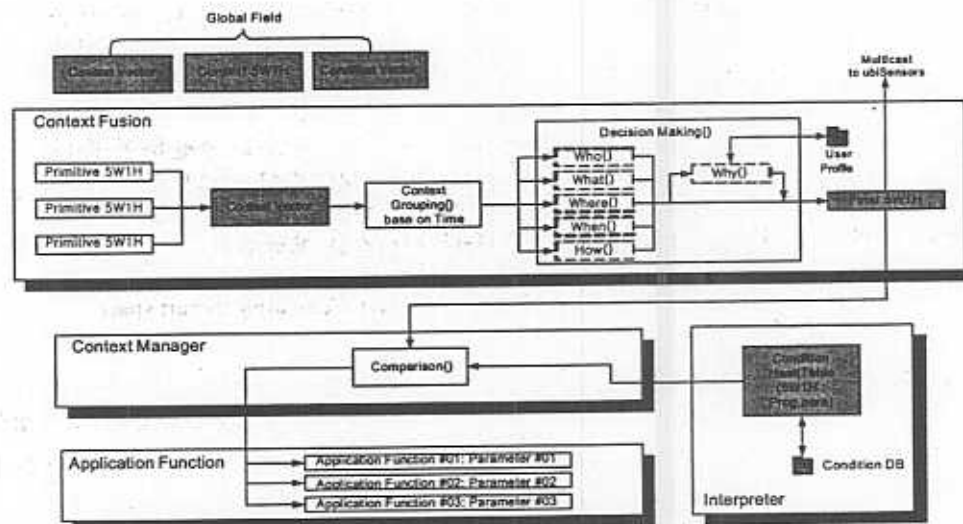


그림 6. 유비서비스를 위한 컨텍스트 패키지