

UCAM2.0: U-city 환경구축을 위한 맥락인식 응용 모델

오유수* · 신춘성** · 서영정*** · 윤효석**** · 한종현***** · 우운택*****

1. 서 론

맥락 인식 시스템은 유비쿼터스 컴퓨팅 분야에서 중요한 연구 주제로 다루어져 왔다[1]. 많은 연구 과제에서 다양한 종류의 센서와 그로부터 파생된 맥락 정보를 활용하여 맥락 인식 시스템을 개발해왔다. 맥락 인식 시스템은 맥락 정보를 생성하고 관리하는 맥락 인식 아키텍처에 의해서 동작된다. 맥락 인식 아키텍처는 사용자의 정보나 주변 환경 변화에 대한 정보, 또는 주어진 상황에 대한 정보를 맥락 정보로 표현하고, 이와 같이 정형화된 맥락 정보를 수집 및 관리하여 현재 상황을 판단하는 일종의 미들웨어이다.

지금까지 스마트 홈/스마트 오피스/스마트 빌

딩 등 주어진 도메인을 위한 맥락 인식 아키텍처에 관한 연구가 활발하게 진행되어 왔다. 그러나 도시나 지역과 같은 규모가 큰 환경이나 이질적인 환경을 위한 맥락 인식 아키텍처에 관한 연구는 아직까지 부족한 상황이다. 특히, 유비쿼터스 컴퓨팅 환경의 도메인을 도시 수준의 규모에 적용한 U-city 환경에서 센서나 서비스들과 같은 다양한 요소들 간의 협업을 위한 일반화된 맥락 인식 아키텍처에 대한 개발이 미흡한 실정이다.

스마트 홈/스마트 오피스/스마트 빌딩 등의 환경에서의 맥락 인식 아키텍처에 관한 연구 사례들은 맥락 인식 아키텍처를 이용한 서비스를 개발해왔다. 특히, 스마트 홈 서비스에 대한 연구 활동들이 활발하게 이루어졌다. EasyLiving (MS Research, since 1998)[2], AwareHome (GATECH, since 1999)[3], Adaptive House (Colorado Univ., since 1999)[4], House_n Project (MIT, since 2000)[5] 가 스마트 홈 연구 동향 중의 일례이다. 이와 같은 스마트 홈 연구 활동들은 맥락 인식 아키텍처를 이용하여 거주자들의 관심이나 의도를 파악하고 다양한 홈 응용 서비스를 거주자들에게 제공한다.

맥락 인식 서비스는 맥락 인식 시스템에서 맥락 인식 아키텍처에 의한 맥락 정보의 처리를 통하여 동작되며, 현재까지 유비쿼터스 컴퓨팅 분야

* 교신저자(Corresponding Author): 우운택, 주소: 광주광역시 북구 오룡동 1번지 광주과학기술원 정보통신공학과 C-308(500-712), 전화: 062)970-2226, FAX: 062)970-3165, E-mail: wwoo@gist.ac.kr

* 광주과학기술원 정보기전공학부 (E-mail: yoh@gist.ac.kr)

** 광주과학기술원 정보기전공학부 (E-mail: cshin@gist.ac.kr)

*** 광주과학기술원 정보기전공학부 (E-mail: ysuh@gist.ac.kr)

**** 광주과학기술원 정보기전공학부 (E-mail: hyoon@gist.ac.kr)

***** 광주과학기술원 정보기전공학부 (E-mail: jhan@gist.ac.kr)

***** 광주과학기술원 U-VR 연구실

* 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장 동력핵심기술개발사업의 일환으로 수행하였음. [2005-S-069-02, 컨텍스트 정형화 및 통합 기술 개발]

에서 핵심 연구 주제로 다루어지고 있다[6]. 특히, 맥락 인식 시스템을 지원하는 아키텍처 및 미들웨어 개발은 맥락 인식 서비스 연구와 함께 진행되고 있다. 특히, Context-Toolkit[7], TEA context acquisition architecture[8], SOCAM[9], JCAF middleware[10], CAMidO[11], 그리고 UCAM 1.0[12] 등이 그것이다. 현재 관련 연구 분야에서는 다양한 센서들을 이용하고 서로 다른 맥락 정보들을 활용하는 맥락 인식 시스템들이 개발되고 있다. 맥락 인식 아키텍처에 대한 이전 연구 결과인 UCAM1.0은 주어진 환경 도메인에서 간단한 맥락 표현, 통합 및 관리 구조를 가지며, 주어진 환경 간의 상호작용을 지원한다[12].

그러나 다양한 환경 도메인에 적용 가능한 일반화된 맥락 인식 아키텍처 개발과 관련된 이슈들은 여전히 풀어야 할 숙제로 남아있다. U-city와 같이 규모가 큰 환경에서 다수 사용자들(군집)의 선호도, 관심이나 의도를 파악하고 사용자 군집 속 개개인에게 개인화된 서비스를 제공하거나 다수 사용자들 간의 동시 서비스 점유 및 이용 충돌을 해결하는 이슈는 맥락 인식 아키텍처를 설계함으로써 논의되어야 한다. 특히, 이전 연구 결과인 UCAM1.0은 고차원적인 맥락 인식 및 커뮤니티 관리, 서비스 프로파일 관리에 대한 지원이 아키텍처 레벨에서 해결되어야 한다.

기존 연구들에 대한 분석으로 다음과 같은 사항들이 U-city 환경구축의 소프트웨어 기반을 위한 맥락 인식 아키텍처 설계 시에 고려되어야 한다.

- 효율적인 시스템 관리를 위하여 아키텍처에서 맥락 정보의 흐름이 명확해야 한다.
- 맥락 원천 요소(센서나 서비스)들 간의 유기적인 관계(네트워크)가 형성되도록 해야 한다.
- 아키텍처 동작시간 중에 맥락 정보들의 추

가/삭제/수정 등의 일상 관리가 가능하여야 한다.

- 다양한 맥락 인식 시스템에서 사용되는 데이터(맥락)들이 아키텍처에서 동기화되어야 한다.
- 이질적인 환경에서의 통신을 위하여 정형화된 맥락 모델을 사용해야 한다.
- 환경의 지능적인 관리를 위해 맥락 정보 통합, 관리 및 추론 기능이 추가되어야 한다.
- 모든 사용자에게 쉽고 널리 사용될 수 있어야 한다.

다수의 사용자들이 생활하는 U-city 환경 구축을 위해서는 위의 요소들을 고려하여 맥락 인식 아키텍처를 설계해야 한다. 이외에도 다양한 센서로부터 입력되는 맥락 정보 획득 방법, 다양한 맥락 정보의 사용자 중심 통합 및 추론 방법 등의 문제들이 명확히 다루어져야 한다.

다양한 센서를 이용하여 맥락 정보를 획득하는 연구는 맥락 인식 컴퓨팅 분야에서 나타난다[13, 14]. 이와 같은 연구는 맥락 정보를 획득하는 서로 다른 보조 시스템을 위한 적합한 인터페이스를 제공할 필요가 있다. 관련 연구 분야에서 맥락 정보를 표현하기 위한 많은 노력이 있었으며, 그 중에서 온톨로지는 맥락 정보를 표현하는 널리 알려진 기법이다. 그러나 정형화된 맥락 정보의 구조나 형식의 정의는 인공지능 분야에서도 여전히 문제점으로 남아있어서 맥락 인식 컴퓨팅 분야에서도 해결해야 할 과제이다. 강건한 맥락 인식 아키텍처는 사용될 센서와 서비스를 충분히 지원하는 맥락 표현법이 있어야 한다. 즉, 새로운 센서나 서비스가 주어진 환경에 추가되어도 맥락 인식 아키텍처가 그 변화를 수용할 수 있는 정형화된 맥락 표현법에 대한 설계가 이루어져야 한다.

맥락 정보 융합 및 추론은 다양한 소스 및 도메

인에 대한 맥락 정보의 통합을 의미한다. 맥락 정보 융합 및 추론은 입력이 추출된 센서 값이 아니라 사전에 계산된 맥락 정보이기 때문에 고수준 맥락 인식으로 간주될 수 있다. U-city 환경에서 협업 환경을 구축하기 위해서는 맥락 인식 아키텍처에서 커뮤니티를 구성하고 관리하는 기법이 필요하다. 그리고 시스템 개발자들이 맥락 인식 아키텍처를 이용하여 맥락 인식 시스템을 쉽게 개발하고 관리하기 위해서는 시스템 개발자들이 시스템의 동작 결과에 대해서 이해해야한다 이와 같은 시스템에 대한 이해 및 개발은 맥락 인식 아키텍처의 서비스 프로파일 관리를 통해서 가능하다.

본 논문에서는 U-city 환경구축을 위한 맥락 인식 응용 모델을 제안한다. 제안된 맥락 인식 응용 모델은 U-city의 다양한 환경에 적용 가능한 일반화된 맥락 인식 소프트웨어 아키텍처이다. 제안된 모델은 아키텍처의 지능을 높임으로써 맥락 인식 능력을 강건하게 한다. 맥락 인식 아키텍처의 지능은 맥락 정보의 의미론적 표현, 해석, 통합, 추론 기능을 통하여 높여질 수 있다. 제안된 모델은 U-city의 효율적인 구성 및 관리를 위하여 U-city 환경 구성 요소들을 사용자 중심의 커뮤니티 단위로 구성하고 관리한다. 사용자 중심의 커뮤니티는 각 관련 요소들을 사용자 중심으로 통합하여 그 사회적 관계를 맥락 정보로 활용한다. 그리고 제안된 모델은 서비스 개발자가 맥락 인식 응용 서비스 개발을 쉽고 빠르게 한다. 즉, 제안된 모델은 서비스 프로파일 관리 및 제공으로 다양한 응용 서비스에 쉽게 적용될 수 있도록 설계되었다.

본 논문은 다음과 같이 구성된다. 2장에서는 맥락 인식 아키텍처 관련 연구들을 분석한다. 3장에서는 제안된 맥락 인식 응용 모델(UCAM2.0)을

소개한다. 4장은 제안된 모델의 특징에 대하여 기술한다. 그리고 5장에서는 제안된 모델이 구현된 어플리케이션과 응용 시나리오를 설명한다. 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

맥락 인식 서비스는 유비쿼터스 컴퓨팅 분야에서 중요한 연구 주제이며, 이를 지원하는 아키텍처 및 미들웨어에 대한 연구로는 Context-Toolkit[7], TEA context acquisition architecture [8], SOCAM[9], JCAF middleware[10], CAMidO [11], 그리고 UCAM1.0[12] 등이 있다. 본 논문에서는 이들 중 몇 가지 최근 연구 동향에 대하여 기술한다.

2.1 SOCAM

SOCAM(a Service-Oriented Context-Aware Middleware)은 온톨로지 기반의 맥락 모델을 사용하고 OSGi (The Open Service Gateway Initiative) 위에서 구동된다. SOCAM의 구조는 독립적인 서비스들로 구성이 된다. 각 서비스는 Context Discovery, Acquisition, Interpretation의 기능을 수행한다. SOCAM은 그림 1과 같이 구성되어있다. Context Interpreter는 맥락 정보를 처리할 수 있도록 논리적 추론 기능(Logic reasoning service)을 제공하며 낮은 수준의 맥락을 높은 수준의 맥락으로 해석해서 제공한다. Context interpreter는 Context reasoner와 Context knowledge base로 구성이 되며, 맥락 추론, 지식 베이스의 일관성을 관리한다. Context database는 맥락 온톨로지와 특정 서브도메인의 이전 맥락을 저장하고 있다. 각 도메인 당 하나의 논리적 맥락 데이터베이스를 지니고 있다.

Context-aware application은 다른 레벨의 맥락을 사용하고 현재의 맥락에 맞게 적응한다. 응용 개발자는 규칙(rule)을 미리 정의하고 파일에 저장하여 context reasoner를 통해 불러올 수 있고, 규칙(rule) 파일을 실행 중에 수정할 수 있다. Service-locating service는 Context provider와 Context interpreter가 자신의 존재를 알릴 수 있게 하고, 다른 사용자와 응용프로그램이 찾을 수 있도록 한다. 그림 1은 SOCAM의 아키텍처이다.

SOCAM은 공통적인 맥락 모델을 사용함으로써 맥락 추론과 지식 공유를 쉽게 하고 어플리케이션 프로토타입 개발을 빠르게 할 수 있다. 또한 SOCAM은 Java를 기반으로 하기 때문에 플랫폼 독립적이며 어플리케이션에게 필요한 서비스의 설치, 활성화, 비활성화, 보완, 제거 등의 기능을 제공하기 때문에 서비스 지향적 어플리케이션에 효과적이다. SOCAM은 다양한 시스템 보안 레벨을 제공한다. 그러나 SOCAM은 Context interpreter와 Context providers의 역할 구분이 모호하며 명확한 모듈의 정의가 부족하다. SOCAM은 온톨로지 추론을 위해서 OWL에 의존하기 때문에 응용 서비스 개발자가 OWL을 이용해야만 하는 번거로움이 있다. 또한, OSGi에 기반을 둔 SOCAM은 OSGi가 기본적으로 해결할 수 없는 보안 문제를 여전히 문제점으로 지니고 있다.

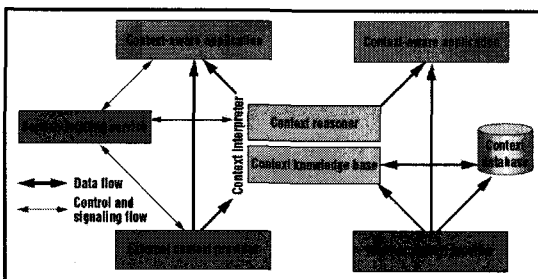


그림 1. SOCAM 아키텍처

2.2 JCAF

JCAF(Java Context Awareness Framework)은 맥락 인식 애플리케이션 개발을 지원하는 프레임워크이다. JCAF는 Context Client, Context Service, Context Sensor and Actuator의 3단 레이어로 구성되어 있다. Context Client Layer는 entity와 맥락에 접근하는 맥락 인식 어플리케이션이다. Context Service Layer의 entity는 맥락 정보의 변화에 반응하게 되며, subscriber를 관리하며 맥락 정보의 이벤트 변화를 client 레이어에 통지하는 역할을 하게 된다. Context service layer의 Transformer Repository는 일반적인 자원들을 유지하고 있으며 적절한 형태로 변형시켜 요청에 응답할 수 있게 한다. Context Sensor and Actuator의 Monitor는 맥락 정보를 센서를 통해 얻어오는 역할을 하며, actuator는 하나 이상의 actuator와 함께 맥락 정보에 변화를 주거나 영향을 미치는 역할을 수행한다. Context Service layer에 전달하는 방식으로는 asynchronous/synchronous 모드의 두 가지가 존재하는데, 첫 번째 방식의 경우 꾸준히 맥락 정보를 전송하는 반면, 두 번째 방식의 경우는 Context Service layer에서 요청이 오는 경우에만 맥락 정보를 전달한다. 그림 2는 JCAF의 아키텍처이다.

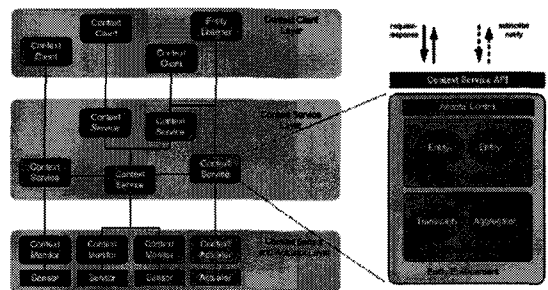


그림 2. JCAF 아키텍처

JCAF는 일반적인 목적, 이벤트 기반 그리고 분산 서비스 지향적인 자바 프레임워크이다. JCAF는 환경 모니터링 센서를 다루는 Context Monitor와 변화에 반응할 수 있는 actuator를 다루는 Context Actuator를 가지고 있다. JCAF는 맥락은 맥락 아이템을 담은 컨테이너로 고려하고 있으며, 맥락의 개체는 맥락 아이템들 중 하나로 해석한다. 그러나 JCAF는 제안된 아키텍처를 몇 가지 프로젝트에 적용해보고 그 결과만을 기술하여 그 효율성을 입증하기가 어렵다. 또한, JCAF의 장점들은 대부분 Java언어가 갖고 있는 특징들이기 때문에 동기화 문제와 같은 단점 역시 지니고 있다.

2.3 CAMidO

CAMidO(a Context-Aware Middleware based on an Ontology meta-model)는 맥락 인식 기능이 추가된 컴포넌트 기반 미들웨어의 확장이다. CAMidO에서 어플리케이션 개발자들은 맥락 수집을 위한 센서들을 다루는 방법, 상위 레벨의 맥락 정보를 추론하기 위한 맥락 해석 등에 대한 코드를 작성해야 한다. CAMidO는 센서와의 통신 그리고 맥락 정보의 수집, 관련된 맥락 감지를 위한 해석과 분석 등이 가능하다. 또한, CAMidO는 온톨로지 메타 모델을 포함하고 있어 어플리케이션 개발자들에 의한 코드 작성을 쉽게 한다. 그림 3은 CAMidO의 아키텍처이다.

CAMidO는 온톨로지 메타 모델을 사용하여 맥락 정보를 기술함으로써 어플리케이션 개발자의 맥락 인식 응용 어플리케이션 제작을 가능케 한다. CAMidO는 기술된 메타 정보를 토대로 맥락 정보를 수집하고 어플리케이션에 적합한 형태로 동작하게 함으로써 개발자가 세부적인 부분에 대한 프로그래밍을 하지 않아도 된다는 장점이 있다.

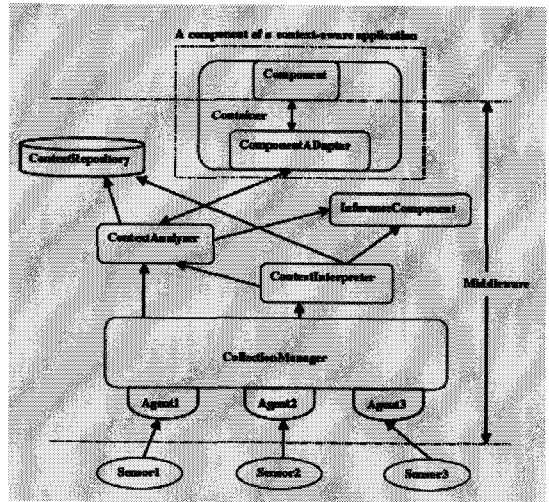


그림 3. CAMidO 아키텍처

CAMidO는 온톨로지 메타 모델을 사용하여 새로운 센서에 대한 맥락 정보를 추가하고, 적응 규칙(adaptation rule)과 해석 정책(interpretation policy)도 추가할 수 있다. 그러나 CAMidO를 사용하면 메타 정보를 작성함으로써 직접적인 프로그래밍은 줄어들고, 소스 코드가 생성된다는 장점이 있지만, 메타 정보를 작성할 때 사용되는 OWL이라는 언어에 대한 지식이 필요하고, 적응 규칙과 해석 정책 작성에도 개발자가 깊숙이 관여해야 한다는 단점이 있다.

2.4 관련 연구와 비교

앞서 언급된 관련 연구들과 제안된 맥락 인식 아키텍처와의 비교는 구조적 특징 및 장점/단점에 대하여 분석함으로써 표현될 수 있다. 표 1에서는 관련 연구들에 대하여 비교한 결과가 정리되어 있다.

표 2는 제안된 맥락 인식 아키텍처인 UCAM 2.0(a Unified Context-aware Application Model 2.0)과 관련 연구들을 맥락 정보의 표현, 해석, 추론, 커뮤니티 관리 및 서비스 제공 등의 기능적인

표 1. UCAM2.0과 관련된 연구들의 비교

| | 구조적 특징 | 장 점 | 단 점 |
|--|---|---|--|
| SOCAM (A Service-Oriented Context-Aware Middleware, 2004) | <ul style="list-style-type: none"> ▶ 독립적인 서비스 구조 ▶ 온톨로지(OWL) 기반 맥락 모델링 | <ul style="list-style-type: none"> ▶ 서비스 지향적 응용 개발에 효율적 ▶ 다양한 맥락 정보 추론 ▶ Java2 Security 모델 채택 | <ul style="list-style-type: none"> ▶ OWL에 의존하여 개발이 어려움 |
| JCAF (The Java Context Awareness Framework, 2005) | <ul style="list-style-type: none"> ▶ 목적, 이벤트 기반 구조 ▶ 분산 서비스 지향적인 Java framework | <ul style="list-style-type: none"> ▶ Java Security API를 이용한 보안 ▶ 실시간 변화에 적응 | <ul style="list-style-type: none"> ▶ 분산 시스템의 문제점 (데이터 동기화) ▶ 네트워크 상의 맥락 서비스 관계 정의 모호 |
| CAMidO (a Context-Aware Middleware based on an Ontology meta-model, 2006) | <ul style="list-style-type: none"> ▶ 온톨로지 메타 모델 ▶ 컴포넌트 기반 구조 ▶ 적응 규칙과 해석 정책 반영 | <ul style="list-style-type: none"> ▶ 추론 규칙 파일과 제어 소스코드 생성을 위한 컴파일러 제공 ▶ 실시간 변화에 적응 | <ul style="list-style-type: none"> ▶ 구조적 흐름 복잡 ▶ 맥락 정보간의 관계 정의 미흡 |
| UCAM1.0 (a Unified Context-aware Application Model, 2006) | <ul style="list-style-type: none"> ▶ 5W1H의 정형화된 맥락 표현 ▶ 센서와 서비스의 독립적인 구조 | <ul style="list-style-type: none"> ▶ 사용자 중심의 정형화된 맥락 정보 표현 및 관리 ▶ 다양한 맥락 정보 통합 ▶ 다양한 센서 및 서비스 응용 개발에 효율적 | <ul style="list-style-type: none"> ▶ 단순한 추론기능 (저수준의 지능) |

표 2. 제안된 UCAM2.0과 관련 연구들과의 기능적 비교

| | SOCAM | JCAF | CAMidO | UCAM1.0 | UCAM2.0 |
|--------|----------------------|-------------------------------|-------------------------------|-----------------------|--|
| 추론 | 다양한 추론, 실시간 추론 규칙 수정 | 맥락 감지에 의한 단순 추론, 실시간 추론 규칙 수정 | 적용적 규칙 생성, 실시간 추론 규칙 수정 | 단순 추론 연산 | 적용적 규칙 생성, 실시간 규칙 수정 |
| 해석 | 논리적 추론 기반 해석 | 엔티티에 의한 구조적 해석 | 해석 정책 기반 해석 | 맥락 정보 해석 기법 | 맥락 정보 해석 기법 |
| 맥락 표현 | 온톨로지 (OWL) 기반 | 객체지향 (J2SE) 표현 | 온톨로지(OWL) 기반 | 객체지향 사용자 중심 표현 (5W1H) | 온톨로지 기반 사용자 중심 의미론적 표현(5W1H) |
| 서비스 제공 | X (서비스 위치 서비스) | X (맥락 클라이언트에 의한 서비스) | X (Container architecture) | 개발자/사용자를 위한 서비스 제공 | 개발자/사용자를 위한 서비스 제공, 빠르고 쉬운 응용 개발을 위한 서비스 프로파일 제공 |
| 커뮤니티 | X | X | X | X | 커뮤니티 구성 및 관리 기법 |
| 구현 도메인 | 실제 환경 (또는 사용자 환경) | 실제 환경 (또는 사용자 환경) | 실제 환경 (또는 사용자 환경) | 실제 환경, 가상 환경, 사용자 환경 | 실제 환경, 가상 환경, 사용자 환경 |

측면에서 비교한 결과이다. 제안된 맥락 인식 아키텍처인 UCAM2.0은 기존 연구에 비해서 맥락 정보의 의미적 표현, 해석, 통합 및 추론을 통하여 아키텍처의 지능을 향상시켰다. 또한, 제안된 UCAM2.0은 커뮤니티 구성 및 관리기법으로 U-city와 같이 규모가 큰 환경에서의 다수 사용자들을 대상으로 하였고, 개발자와 사용자 모두에게 쉽게 맥락 인식 응용 서비스를 개발하거나 사용할 수 있도록 설계되었다.

3. 정형화된 맥락 인식 응용 모델: UCAM 2.0

3.1 UCAM2.0 아키텍처

본 논문에서는 다양한 센서나 서비스들 간의 협업 공간인 U-city 환경 구축을 위한 일반화된 맥락 인식 아키텍처를 제안하였다. 제안된 아키텍처는 사용자와 현실세계, 가상세계의 이질적인 환경 구성 요소(오브젝트)들 사이의 이음매 없는 연결을 지원하는 정형화된 맥락 인식 응용 모델(UCAM2.0: a Unified Context-aware Application Model)이다.

제안된 UCAM2.0은 UCAM1.0의 확장 버전으로 센서와 서비스의 독립적인 구조를 갖는다 [12,15]. UCAM2.0의 센서 구조는 신호 감지/특징점 추출/초별 맥락 생성의 과정을 거치는 UCAM1.0의 센서 구조와 동일하다. 정형화된 맥락 인식 응용 모델인 UCAM2.0의 서비스 아키텍처는 그림 4와 같다. 각 모듈 간의 관계 및 맥락 흐름은 이전 연구에 정의되어 있다 [12,15]. UCAM 센서는 사용자와 그 주변 환경에 대한 변화를 인식하여 초별 맥락 정보(Preliminary Context)를 형성하는 역할을 한다. 그리고 UCAM 서비스는 수집된 맥락 정보들을 인식하

여 사용자의 의도에 따른 응용 서비스를 제공하는 역할을 한다.

그림 4에서 보는 것과 같이, UCAM2.0의 Communicator는 U-city의 효율적인 구성 및 관리를 위하여 U-city 환경 구성 요소들을 커뮤니티 단위로 구성하고 관리한다. UCAM2.0의 Context Integrator/Context Manager/Interpreter는 정형화된 맥락 표현 모델을 채택하여 맥락 정보 통합, 추론, 관리, 해석의 정보 전달 과정을 체계적으로 수행한다. 그리고 제안된 UCAM2.0은 Service Provider의 서비스 프로파일 관리 및 제공 기능으로 다양한 응용 서비스에 쉽게 적용될 수 있다.

제안된 UCAM2.0은 맥락 정보의 의미적 표현, 해석, 추론과 같은 의미론적인 지능을 지원한다. 제안된 아키텍처는 맥락 인식 능력 향상을 위해서 다양한 인공지능 알고리즘을 애드온(add-on) 기능으로 쉽게 연결할 수 있다. 그리고 제안된 아키텍처는 5WIH (Who, What, Where, When, How, and Why)의 육하원칙에 의한 맥락 정보 표현 방식(Context Model)에 기반을 두어 맥락 정보들을 관리한다. 사용된 맥락 정보 표현 방식(Context Model)은 맥락 저장소(Context Repository)의 역할을 하며 맥락 정보의 히스토리를 관리한다. 제

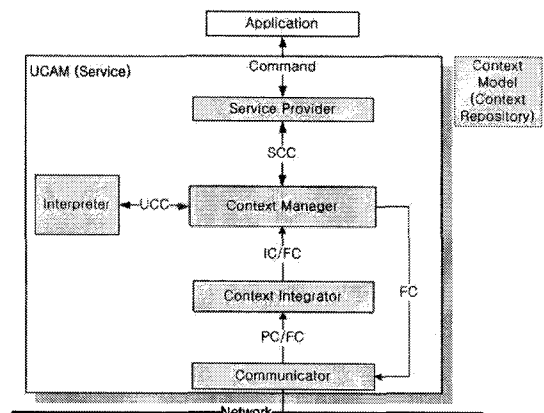


그림 4. UCAM2.0 아키텍처 (UCAM 서비스 부분)

안된 아키텍처는 사용자, 실제 환경 및 가상 환경의 오브젝트 사이의 관계를 규명하고 관리하는 커뮤니티 관리 기능을 가지고 있다. 마지막으로 제안된 아키텍처는 개발자 및 사용자들이 쉽게 사용할 수 있도록 공개 소스 정책을 가지고 있으며 서비스 프로파일을 통하여 어떤 맥락 인식 응용 서비스에도 쉽고 빠르게 개발될 수 있다.

3.2 맥락 정보 표현

UCAM2.0의 맥락 표현 방식(Context Model)은 온톨로지 기반의 구조와 추론 연산을 지원하는 사용자 맥락 정보의 집합으로 정의된다. 맥락 표현 방식은 UCAM2.0의 각 모듈의 연산을 돕고 맥락 정보의 저장 관리를 가능하게 하며, 사용자 중심의 맥락 정의와 계층적 구조를 가진다. 사용자 중심의 맥락 표현 방식(Context Model)은 5W1H 형태로 구조화한 맥락 요소(ContextElement), 맥락 요소들을 처리할 수 있는 연산자를 지닌 맥락(Context), 그리고 맥락 정보를 저장하고 검색할 수 있는 맥락 메모리(ContextMemory)로 구성된다[16].

맥락 정보의 표현은 맥락 인식 아키텍처 설계시에 필요한 요소 중의 하나이다. 5W1H 표현법은 맥락 정보가 누가(Who), 언제(When), 어디서(Where), 무엇을(What), 어떻게(How), 왜(Why)의 육하원칙으로 표현되는 구조화된 형식이다. 5W1H의 표현법은 정형화된 맥락 인식 응용 모형에 적합한 맥락 정보를 사용자 중심의 관점으로 제공한다. 이 표현법은 육하원칙을 이용하여 수집된 정보를 구조화하므로 센서에 의하여 검출될 수 있는 사용자의 상황을 정확하게 표현할 수 있다.

또한, 이 맥락 표현은 각각의 서비스에 따른 맥락 정보의 제공 및 관리를 쉽게 한다. 각 5W1H

맥락 정보는 계층적으로 구조화되어 있다. '누가(Who)'의 맥락은 거주자의 신원, 특징 및 관계성에 대하여 기술한다. '언제(When)'의 맥락은 센서 또는 서비스에 의해서 생성되는 시간 정보를 나타낸다. '어디서(Where)'의 맥락은 거주자의 실내 위치와 실외 위치에 대하여 기술한다. '무엇을(What)'의 맥락은 거주자에 의해서 사용되는 센서 혹은 서비스의 정보이다. '어떻게(How)'의 맥락은 거주자의 몸 상태, 제스처, 또는 제어 명령 등에 대한 내용을 기술한다. 마지막으로 '왜(Why)'의 맥락은 수집되는 정확한 데이터로부터 추론되는 정보이며, 거주자의 의도 및 감정에 대해서 기술한다[17].

4. UCAM2.0의 특징

UCAM2.0의 특징으로는 맥락 융합 및 추론, 해석, 학습의 맥락 인식 기능과 사용자 중심 커뮤니티 관리 기능, 그리고 서비스 프로파일 관리 기능의 세 가지로 나뉜다.

4.1 맥락 인식

4.1.1 맥락 융합 및 추론

맥락 정보 융합 및 추론은 제안된 맥락 인식 응용 모델에서 맥락 통합기에 의해서 제공되는 핵심 기술 요소이다. 맥락 통합기는 다양한 종류의 센서로부터 획득된 센서 데이터를 융합하고 서로 다른 수준의 정보를 통합하도록 설계되었다. 이것은 의미적인 맥락 융합을 하는 시스템에서 결과를 만든다. 하위 레벨의 데이터의 융합은 센서 내부의 요소로 고려된다. 맥락 통합기는 사용자 중심의 맥락 정보로써 거주자(특히, Who의 맥락)에 따른 분류와 융합을 하도록 지원한다. UCAM2.0에서의 맥락 융합 및 맥락 추론은 저수

준의 맥락 정보들로부터 고수준의 맥락 정보를 이끌어내는 논리적 과정이다. 맥락 추론은 맥락 표현을 사용할 때 맥락 정보의 일관성을 유지시킨다. 맥락 추론은 맥락 히스토리를 이용하여 맥락 정보를 재사용하고, 다양한 환경에 적합한 추론 규칙을 재구성한다. UCAM2.0에서 맥락 추론은 맥락 통합기에서 센서로부터 수집된 다양한 맥락 정보들을 사용자 중심으로 융합하고, 이렇게 융합된 맥락을 이용하여 적절한 인공지능 추론 기법을 거치는 일련의 과정이다.

다양한 종류의 센서로부터 다수의 맥락 정보 입력을 동시에 통합하여 U-city의 각 사용자의 activity에 대한 의도를 추론한다. 각 사용자의 의도 추론은 UCAM의 맥락 통합기에서 이루어진다. 그림 5는 맥락 통합기의 구조를 보여준다.

Context Object Analyzer는 활성영역에 위치한 다양한 종류의 센서들로부터 맥락 정보를 수집하고 ‘누가(Who)’의 부분 맥락 중의 사용자의 신원 정보에 따라서 수집된 맥락 정보들을 분류한다. Context Repository는 맥락 표현 방식(Context Model)의 역할을 의미하며, 통합된 맥락 정보들의 히스토리를 저장하고 관리한다. Preliminary Context Fusion module은 입력된 초벌 맥락 정보들을 각 특성에 따라서 4W1H(Who, What, Where, When, and How)의 통합 맥락으로 구성한다. Context Inference Engine은 Preliminary Context Fusion module의 결과를 이용하여 ‘왜(Why)’의 맥락을 추론해낸다. Integrated Context Generator는 이미 구성된 4W1H의 맥락 정보와 추론된 ‘왜(Why)’의 맥락을 통합하여 5W1H의 통합 맥락을 생성한다. 이와같이 형성된 통합 맥락(IC: Integrated Context)은 사용자 의도 및 감정 정보를 포함한 완전한 5W1H 맥락 형식이다.

맥락 통합기는 맥락 통합 과정에서 특히 맥락의 재사용을 위한 메카니즘으로 구성되었다. 맥락의 재사용은 추론 과정에서 이전 과정의 추론된 결과를 다시 사용하는 연속 과정의 결과물로서 시스템의 효율성을 높일 수 있다. 맥락의 재사용의 과정이 진행될수록 높은 수준의 맥락이 추론될 수 있다. 또한, 맥락의 재사용은 맥락 메모리(맥락 표현)에서 관리되는 맥락 히스토리에서 저장된 맥락 정보를 사용함으로써도 이루어진다.

맥락 통합기에 기반을 둔 맥락 인식 시스템의 일반화된 행동과 기능은 규칙(rule)에 의하여 생성될 수 있다. 개발자나 관리자에 의한 시스템의 변화 및 재구성은 규칙(rule)을 추가, 삭제 및 수정함으로써 가능하다. 이와 같은 규칙(rule)을 간략화하기 위해서 일반적인 텍스트 에디터를 이용하여 쉽게 재구성할 수 있다. 그러므로 맥락 통합기에서 소스 코드를 바꾸지 않고도 전체 시스템 행동에 대한 변화가 가능하다. 맥락 통합기에서 구현된 규칙(rule)은 LISP 언어의 구문에 기반한다. 다른 선언적 프로그래밍 언어와 비교하여 이와 같은 구현은 규칙(rule)의 순서에 상관없이 재구성을 쉽게 한다.

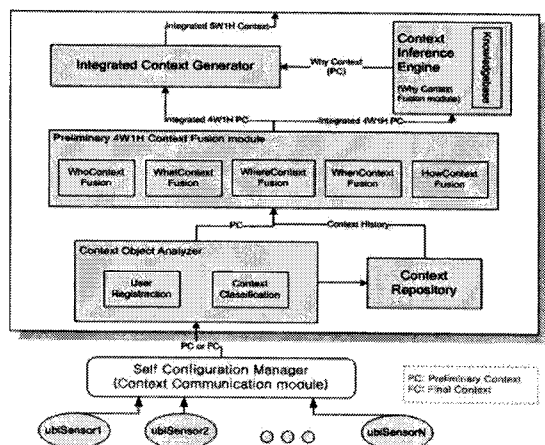


그림 5. 맥락 통합기의 구조

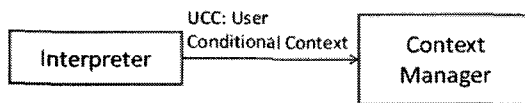
4.1.2 맥락 해석

UCAM2.0에서의 맥락 해석 (Context Interpretation)은 다른 목적을 수행하는 두 요소(요소: UCAM2.0의 각 담당 모듈)간의 상호 운용 및 호환성을 위하여 한 요소의 어휘로 표현된 5WIH의 맥락 정보를 다른 요소의 어휘로 표현된 5WIH의 맥락 정보로 변환하는(translation) 프로세스이다. UCAM2.0에서의 맥락 해석은 Interpreter에서 담당한다. Interpreter에서는 맥락정보를 서비스 파라미터 형태로 변환을 하거나 다른 응용 도메인 간 맥락 요소를 변환한다. 이를 위해서는 맥락 정보에 대한 형식과 서비스 파라미터에 대한 어휘 정보가 필요하다. 그림 6은 UCAM2.0에서 지원되는 세 가지 해석 방법을 보여준다.

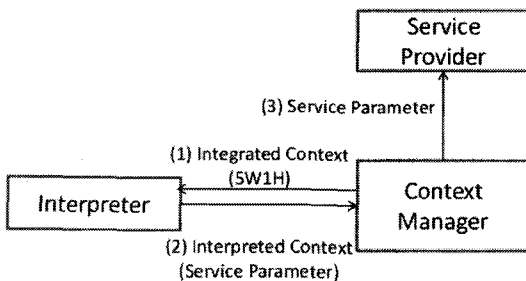
그림 6(a)는 사용자의 어휘로 구성된 사용자 조건 맥락 정보(UCC: User Conditional Context)를 해석하여 Context Manager로 넘겨주는 프로세스를 보여준다. 그림 6(b)는 Context Manager를 통해 통합 맥락 정보를 Interpreter를 통해 Service Provider가 사용하는 서비스 파라미터의 어휘로 변환하는 프로세스를 나타낸다. 그림 6(c)

는 Service Provider에서 사용되는 서비스 파라미터를 Context Manager에서 Interpreter를 통해 5WIH 형태의 맥락 정보로 변환하는 단계를 보여준다. Interpreter는 이와 같이 사용자의 어휘를 맥락 인식 응용 모델에서 사용하는 어휘로 변환을 하거나, 맥락 정보를 서비스 파라미터로 또는 서비스 파라미터를 맥락 정보로 해석을 하는 기능을 한다. 해석을 하기 위해서는 맥락 정보에 사용되는 어휘와 Servicer Provider를 통해 실행되는 응용에 대한 어휘에 대한 정보가 필요하다. UCAM2.0에서는 개발자가 각 서비스를 구현할 때에 응용 서비스에 대한 어휘를 룩업(lookup) 테이블 형태로 내포하도록 권장하며, 맥락 정보에 대한 이해는 5WIH에 대한 형식을 기본으로 가정한다.

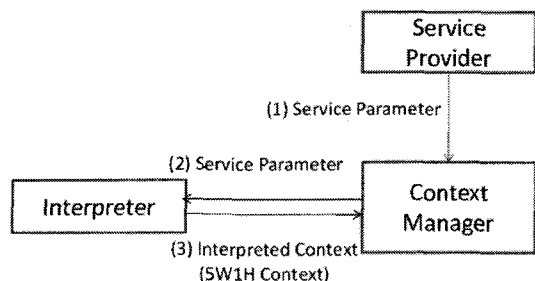
맥락 해석을 하기 위해서 Interpreter는 입력에 대응하는 출력을 결정한다. 예를 들어 Interpreter는 사용자 조건 맥락(UCC)에 사용되는 어휘를 룩업 테이블에서 검색을 하여 이에 대응하는 출력을 결정한다. Interpreter는 입력이 통합 맥락(IC) 형태인지 서비스 파라미터인지에 따라 룩업테이블



(a) 사용자 조건 맥락(UCC) 해석



(b) 맥락 정보-서비스 파라미터 해석



(c) 서비스 파라미터-맥락 정보 해석

그림 6. UCAM의 맥락 해석

에서 출력을 선택하게 된다. 이런 해석 방법을 통해 다른 도메인 간 (예) 사용자-서비스, 영어-한국어, 단위 변환 등) 의 어휘 정보를 변환할 수 있다. UCAM2.0은 맥락 해석을 통하여 여러 유형의 정보가 제공되는 U-city에서 상호 운영 및 호환성을 유지할 수 있도록 한다.

4.1.3 맥락 학습

맥락 인식 어플리케이션이 환경에 편재되어 있는 센서, 사용자 입력, 또는 가상현실 어플리케이션들로부터 맥락 정보를 획득함에 있어서, long-term 맥락 정보 처리에 대한 고려는 매우 중요하다. 이는 환경의 상태 정보, 사용자의 의도 정보, 그리고 가상 아바타의 지능적인 반응 정보를 정확히 평가할 수 있도록 하기 위함이다. 사용자의 현재 시간, 장소, activity와 같은 short-term 맥락 정보와 관심사 및 life-log와 같은 long-term 맥락 정보는 서로 다른 time scale로 학습될 수 있는 구조적 지원이 필요하다. 따라서 맥락 정보의 특성에 따라 적절한 학습 메커니즘을 활용하는 UCAM2.0에서의 맥락 학습은 맥락 인식 어플리케이션 개발을 위해 필수적이며 UCAM2.0의 Context Manager에서 지원된다. 특히, 서비스 사용에 대한 사용자 맥락 히스토리 정보를 파악하기 위해서는 개인별 기본적인 프로필 정보와 함께 서비스 사용에 대한 개인별 성향이나 특징에 해당하는 사용자 맥락 정보가 많은 시간 동안 수집 및 분석되어야 한다. 이 때, 휴대용 정보 단말기는 사용자가 지니고 있는 상태에서 각 서비스 사용에 대한 맥락 조건을 쉽게 분석할 수 있는 사용자의 서비스 사용 피드백 정보를 수집하여 학습함으로써 사용자가 맥락 조건을 직접 지정하지 않는 상황에 대한 서비스 제공도 최적화시킨다. 또한, 개인화된 서비스 제공을 위하여 사용자의 서비스 사용에 대한 선호도 정보의 학습을 통하여 각 사

용자의 개별 선호도 정보를 반영한 서비스 특성 및 내용 제공이 보장된다. 특히, 제안된 맥락 인식 응용 모델에서는 환경에 있는 서비스들로부터 최종 맥락 정보들(FC: Final Context)을 수집하여 관리함으로써, 동적으로 변하는 서비스 사용에 대한 사용자 맥락 정보를 획득할 수 있다. 주기적으로 수집되는 최종 맥락 정보들(FC)을 활용하여 동적 사용자 맥락 정보를 분석하고 학습한다. 결국, 매칭 및 학습 과정 후에, 사용자가 특정 상황에 특정 서비스를 제공받고자 하는 간접적인 의도를 반영하는 사용자 조건 맥락 정보(UCC)를 업데이트함으로써 사용자에게 보다 더 개인화된 서비스를 제공한다.

4.2 사용자 중심의 커뮤니티 관리

UCAM2.0의 Communicator는 UCAM2.0의 모듈들 간의 맥락 정보를 전달하는 기본적인 기능뿐만 아니라 커뮤니티 구성을 통해 U-city 환경에서 협업 환경을 구축하는데 도움을 준다. UCAM2.0에서의 커뮤니티는 사용자의 목적을 중심으로 사용자, 서비스(장치), 그리고 콘텐츠 등이 커뮤니티의 구성원으로 포함될 수 있는 사용자 중심의 커뮤니티를 정의하여 구성한다. 사용자의 목적을 알기 위해 Context Manager로부터 최종 맥락(FC)을 입력 받게 되고 이에 해당하는 커뮤니티 구성원을 결정하게 된다. Communicator는 생성된 맥락 정보를 구성된 커뮤니티 구성원들에게 선택적으로 전달하며 이를 통해 맥락 정보의 불필요한 전달이 없도록 한다.

커뮤니티 구성원을 결정하기 위해 Communicator는 사회 연결망을 구성하게 되는데 UCAM2.0의 모듈들 사이의 관계성을 이용하여 협업 환경 구성에 도움이 되는 구성원을 찾게 된다. 이때 이용되는 관계로는 사용자와 사용자, 서비스와 서

비스, 콘텐츠와 콘텐츠 사이의 관계뿐만 아니라, 사용자와 서비스, 사용자와 콘텐츠, 콘텐츠와 서비스 등 이질적인 개체들 사이의 관계성까지 포함하여 사회 연결망을 구성한다. UCAM의 개체 디스커버리 기능을 통해서 환경에 존재하고 있는 사용자 및 서비스 그리고 센서 등에 대한 상태 정보를 얻게 되고, 사용자의 목적을 달성하기 위해 필요한 센서, 서비스 및 콘텐츠 등을 선택하게 된다.

생성된 커뮤니티는 맥락 정보의 변화에 따라 유지 또는 재구성된다. Communicator는 다른 개체로부터 들어오는 맥락 정보와 Context Manager에서 만들어진 최종 맥락 정보(FC)를 입력으로 받아 개체들 사이의 관계성을 업데이트 하게 되고 사용자의 목적 달성에 긍정적인 역할을 하는 개체를 커뮤니티 구성원으로 추가한다. 개체 디스커버리 기능을 통해 수신한 사용자, 서비스 프로파일 정보를 이용하여 적합한 커뮤니티 구성원을 결정하며, 맥락 정보의 변화에 동적으로 반응하여 커뮤니티의 목적 및 구성원을 갱신하는 작업을 수행하게 된다. 주기적으로 사용자 주변 환경의 맥락 변화를 모니터링 하게 되며, 맥락의 변화가 감지되는 경우 새롭게 얻은 맥락 정보와 기존 커뮤니티의 목적을 비교하여 구성원의 목록을 갱신하는 작업을 수행한다. 그림 7은 앞에서

설명한 커뮤니티를 관리하는 프로세스의 상태 변화를 도식화하여 나타내고 있다. 이러한 프로세스를 통해 커뮤니티 기반의 U-city의 협업 환경을 동적으로 구성하게 된다.

4.3 서비스 프로파일 관리

맥락 인식 서비스는 상황에 따라 사용자가 필요로 하는 서비스를 제공하기 위해 U-city 환경 내의 사용자 및 주변 센서로부터 맥락을 수집 및 분석을 수행할 뿐만 아니라 환경에 있는 다른 서비스와 지속적인 상호작용을 수행한다. 이를 위해 맥락 인식 서비스는 서비스에 대한 기본적인 정보 및 특정한 상황에 반응하기 위한 정보를 필요로 한다. 서비스 프로파일은 서비스가 U-city 환경에서 사용자 및 다른 서비스와 상호작용을 수행하기 위해 필요한 정보를 기술한다. 따라서 맥락 인식 서비스는 사용자에게 디스커버리 되었을 때 서비스가 제공 가능한 서비스 목록을 제공하기 위한 정보를 갖는다. 또한 다른 서비스와 협업을 수행하기 위해 필요한 동작에 대한 상세 정보를 기술한다. 그리고 개발자가 지정한 특정한 상황에서 서비스가 자동으로 반응하도록 하기 위한 기본 서비스 조건을 포함한다. 이러한 서비스 프로파일은 개발자들이 쉽게 설정할 수 있도록 XML을 통해 기술된다. 그림 8은 서비스 프로파일의 예제로써 XML을 기반으로 기술된 TV 서비스 프로파일의 내용을 나타낸다.

그림 8과 같이 TV서비스 프로파일은 서비스에 대한 기본 이름을 가짐과 동시에 볼륨(volume)이라는 서비스 속성(Attribute)을 갖는다. 그림 8에서 볼륨은 정수형 값이며 기본 설정 값으로 5를 갖는다. 또한 위치정보는 TV 서비스가 TV 장치에 설치되어 있으면서 ubiHome이라는 스마트 환경을 서비스 영역으로 설정하고 있음을 나타내고

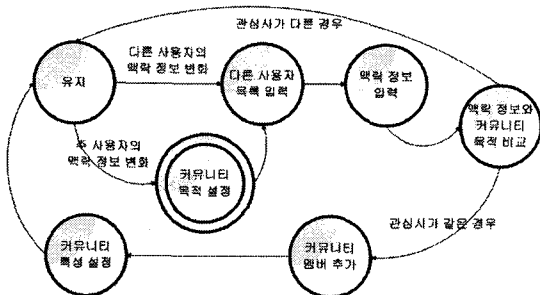


그림 7. 사용자 중심의 커뮤니티 관리 프로세스 상태 변화

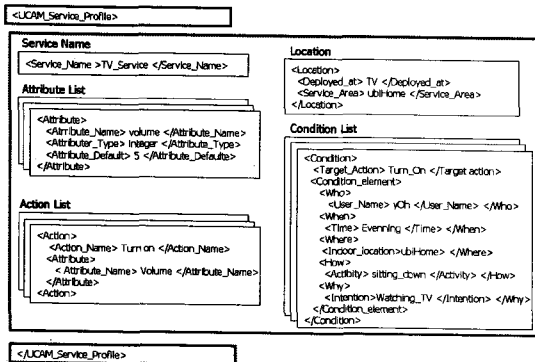


그림 8. XML기반 TV 서비스 프로파일

있다. 그리고 서비스로 제공 가능한 서비스 작용 (Action)으로는 여러 가지가 있으며 그중에서도 Turn_on의 경우 이름 및 관련 속성을 기술하고 있다.

서비스 프로파일은 서비스 실행됨과 동시에 Service Provider를 통해 관리된다. Service Provider는 개발자에 의해 제공되는 기본 정보 및 서비스 작용에 대한 정보를 XML 파일로부터 읽고 이를 관리한다. Service Provider는 Context Manager로부터 최종 맥락(FC)이 수신되면 해당하는 동작을 수행하며, Context Manager 및 Communicator가 최종 맥락(FC)과 커뮤니티를 구성하는데 각각 활용된다. 그림 9는 서비스 실행 및 동작에 따른 서비스 프로파일의 흐름을 나타낸다.

그림 9와 같이 서비스 개발자는 XML 스크립트를 통해서 서비스를 기술한다. Service Provider는 실행과 동시에 개발자가 작성한 XML기반

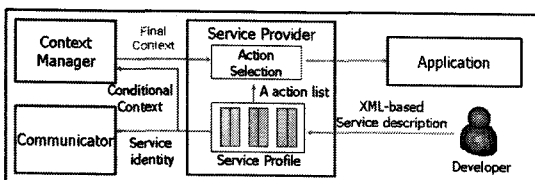


그림 9. 서비스 프로파일 관리

프로파일을 로딩하면서 서비스에 대한 정보를 설정하며, 로딩된 프로파일은 Communicator, Context Manager, 그리고 Service Provider에 제공된다.

5. 맥락 인식 응용 모델의 적용

5.1 맥락 인식 응용 모델의 구현

5.1.1 유비쿼터스 컴퓨팅 환경을 위한 맥락 인식 응용 모델 구현

UCAM2.0의 유용함을 검증하기 위해 제안된 맥락 인식 응용 모델을 환경에 적용한 ubi-UCAM을 중심으로 맥락 인식 미디어 서비스를 구현하였다. 맥락 인식 미디어 서비스는 사용자의 프로파일과 주변 센서로부터 수집된 맥락을 활용하여 사용자에게 개인화된 미디어 콘텐츠를 제공한다. 뿐만 아니라 다수의 사용자가 동시에 미디어 서비스를 이용하는 경우, 이들 간에 발생가능한 충돌을 감지하고 해결하여 다수의 사용자들이 조화롭게 미디어 서비스를 이용하도록 한다.

그림 10과 같이 맥락 인식 미디어 서비스는 사용자의 모바일 장치로부터 사용자 프로파일을 주변 센서로부터 사용자의 주변의 환경에 대한 정보를 수집한다. 그리고 수집된 맥락과 사용자의 프로파일을 분석하여 사용자가 필요로 하는 서비스

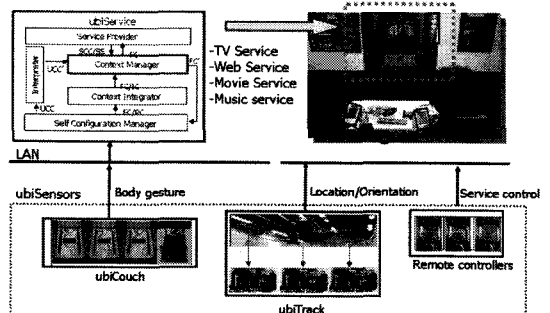


그림 10. 맥락인식 미디어 서비스의 개념도

를 선택한다. 이를 통해 얻은 최종 맥락(FC)을 통해 미디어 서비스는 특정 사용자의 상황에 부합되는 개인화된 미디어 서비스를 제공한다. 또한 다수의 사용자가 동시에 미디어 서비스를 이용하는 경우 이들의 프로파일을 포함한 맥락 정보를 활용하여 충돌을 감지하고 이들의 선호도를 가장 잘 반영한 미디어 아이템을 제공함으로써 충돌을 해결한 미디어 서비스를 제공한다.

5.1.2 착용형 컴퓨팅 환경을 위한 맥락 인식 응용 모델 구현

유비쿼터스 컴퓨팅 환경에서 맥락 인식 서비스 제공을 위한 모델인 ubi-UCAM은 환경에 있는 응용 서비스들로 하여금 사용자 행동 패턴을 지속적으로 모니터링하며, 사용자 개개인의 서비스에 대한 요구 및 선호도를 파악하여 개인화된 서비스를 제공하는데 중점을 둔다. 그러나 각각의 서비스가 관리하는 방식은 서비스가 추가될 때마다 서비스 실행을 위한 조건을 모든 사용자가 각 서비스마다 직접 입력해야 하는 번거로움이 생기고, 시간이 지남에 따라 복잡해지는 사용자 별 서비스 사용에 대한 조건을 관리하는데 어려움이 발생한다. 또한, 한 사용자의 다수의 서비스 사용에 대한 선호도 정보의 분석을 통한 보다 정확하고 깊이 있는 사용자 분석이 어렵다. 따라서 서비스 특성이나 서비스의 구체적 내용이 사용자 별로 적용적으로 반응할 수 있는 개인화된 서비스 제공 방법을 지원하기 위하여 휴대용 정보 단말기가 직접 사용자의 서비스에 대한 선호도를 인식하고 관리하는 착용형 맥락 인식 응용 모델(wear-UCAM)을 ubi-UCAM과는 구별하여 구현하였다. 스마트 홈 환경의 모바일 사용자 인식을 위해서는 사용자 몸에 부착되어 있는 센서들이 사용자에 대한 정보를 휴대용 정보 단말기로 전달하며, 각 정보 단말기는 다양한 센서들로부터 획득한 정보와 정보

단말기를 통해 제공되는 사용자의 직접적인 요구 정보를 서비스 목적에 맞게 재해석하여 사용자 선호도에 맞는 서비스 내용을 제공한다. 여기서 UCAM2.0은 환경에 있는 사용자가 휴대하는 모바일 기기에 구현되는 착용형 맥락 인식 응용 모델(wear-UCAM)과 환경의 응용 서비스에 구현된 맥락 인식 응용 모델(ubi-UCAM)들이 서로 분산된 독립적인 객체로써 존재하여 서비스의 제공 및 공유 등의 상호작용이 이루어지도록 하는 공통 메커니즘을 지원한다.

그림 11에서 볼 수 있듯이, 네트워크 통하여 환경에 있는 서비스들에게 보급된 사용자 조건 맥락 정보들(UCC)은 환경에 있는 서비스들이 사용자가 원하는 해당 서비스를 구동시키기 위하여 활용된다. 즉 사용자의 서비스에 대한 선호도 정보가 환경에 전달됨으로써, 개인화된 서비스 제공이 가능하다. 구현된 모델(wear-UCAM)은 휴대용 정보 단말기를 통해 사용자와 환경과의 상호작용을 필요로 하는 다양한 어플리케이션에 광범위하게 활용될 수 있으며, 사용자의 사적인 정보 보호를 보장하면서 개인화된 서비스 제공을 가능하게 할 수 있다.

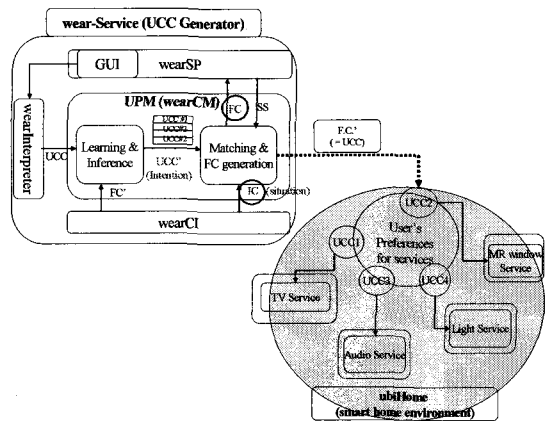


그림 11. wear-UCAM(사용자)과 ubi-UCAM(환경)과의 상호작용 과정

5.2 응용 시나리오

U-city 환경 구축의 소프트웨어 기반을 위한 UCAM2.0을 활용한 응용 시나리오는 U-city에 위치한 레스토랑에서의 메뉴 및 음식 추천 서비스, U-city의 와인 바에서의 와인 추천 서비스, U-city의 대형 할인 마트에서의 음식 재료 추천 서비스, U-city의 헬스클럽에서의 운동 추천 서비스 등이 될 수 있다. 위 시나리오는 사용자 센서 및 주변 환경 센서들로부터 획득된 맥락 정보를 활용하여 사용자에게 가장 적합한 서비스를 추천하는 경우이다. 즉, 위의 다양한 응용 시나리오는 제안된 맥락 인식 아키텍처인 UCAM2.0을 음식/재료/운동 서비스에 적용하여 맥락 인식 추천 서비스를 구현하는 것이다.

U-city의 레스토랑에서의 메뉴 추천 서비스 시나리오는 메뉴판에 부착된 마커나 RFID 태그 등으로부터 감지된 맥락 정보와 착용형 장치를 통한 사용자 프로파일 정보, 그리고 주변 환경 센싱 정보를 UCAM2.0이 수집 및 통합하여 통합된 맥락을 해석한 후에 사용자에게 가장 적합한 메뉴를 추천하는 웰빙 라이프 시나리오이다. 만약 일행이 있을 경우, UCAM2.0은 구성원 모두의 사용자 프로파일 정보 및 센서 정보로 인하여 음식 성향에 관련된 커뮤니티를 형성하고 구성원 모두에게 적합한 메뉴를 추천하도록 한다.

U-city의 와인 바에서의 와인 추천 서비스 시나리오는 와인 병에 부착된 마커나 태그로부터 정보를 획득하고 와인 생산년도, 맛(떫은 맛, 달콤한 맛 등)에 대한 사용자의 와인 취향 프로파일 정보와 그 때의 상황에 따라 감지된 주변 센싱 정보를 이용한다. 이외에도 식품이나 운동 등의 추천 서비스들도 UCAM2.0이 적용되어 사용자에게 적용될 수 있다. 이와 같은 시나리오에서 알 수 있듯이, UCAM2.0은 음식/재료/운동 서비스를

위한 장소들이 서로 다르기 때문에 동일 사용자에게 적합한 서비스 제공을 위해서는 서로 네트워크로 연동이 되어있어야 하며, 공통된 맥락 모델을 통하여 이질적인 환경 간의 통신이 가능하다. 또한, UCAM2.0은 일반화된 맥락 인식 아키텍처이므로 시각/청각/촉각 등을 위한 증강현실 기법과도 연동이 가능하여 실감형 서비스를 사용자에게 제공할 수 있다.

6. 결 론

본 논문에서는 U-city 환경구축의 소프트웨어 기반을 위한 맥락인식 응용 모델(UCAM2.0)을 제안하였다. 제안된 UCAM2.0은 사용자와 현실세계, 가상세계의 이질적인 환경 구성 요소(오브젝트)들 사이의 이음매 없는 연결을 지원하는 일반화된 맥락 인식 아키텍처이다. 제안된 모델은 정형화된 맥락 표현 및 센서와 서비스의 독립적인 구조로 인하여 이질적인 환경 간의 동적 변화에 적응 가능한 아키텍처이다. 제안된 UCAM2.0은 U-city 환경의 일상적인 관리 및 유지를 위하여 맥락 인식 아키텍처의 지능을 향상시킨다. 제안된 UCAM2.0은 정형화된 맥락 모델을 채택하여 맥락 정보 해석, 통합, 추론, 관리의 정보 전달 과정을 체계적으로 수행한다. 또한 제안된 UCAM2.0은 U-city와 같이 규모가 크고 사용자, 실제 환경 및 가상 환경과 같은 이질적인 환경을 포함하는 조건에서 각 환경의 구성요소들 사이의 관계를 규명하고 커뮤니티를 구성하여 관리한다. 그리고 제안된 UCAM2.0은 서비스 프로파일 관리 및 제공으로 다양한 응용 서비스에 쉽게 적용될 수 있도록 설계되었다.

제안된 맥락 인식 응용 모델의 효용성은 유비쿼터스 컴퓨팅 환경과 착용형 컴퓨팅 환경에서 구현된 예제를 통해서 평가받을 수 있으며, U-

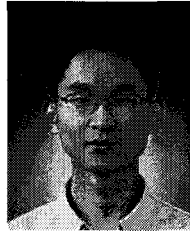
city에서 UCAM2.0을 활용한 다양한 응용 시나리오를 통해서 실생활에서의 파급효과를 기대할 수 있다. 추후 논의되어야 할 사항으로는 사용자의 행동을 예측하는 방법과 현실 및 가상 환경에서의 서로 다른 오브젝트와의 통신 방법이 고려되어야 한다. 그리고 맥락 인식 응용 서비스를 실생활에서 자유롭게 사용하기 위해서는 환경 구성 요소들의 자유로운 추가/삭제/수정 등의 실생활 관리 기법이 맥락 인식 응용 모델에 마련되어야 한다. 마지막으로 설계된 맥락 인식 응용 모델이 U-city 환경에 적합한가를 판단하는 평가 기준을 정립하여 실제 환경에서의 필드 테스트가 이루어져야 한다.

참 고 문 헌

- [1] Schilit, B., Adams, N., & Want, R., "Context-Aware Computing Applications," *In Proceedings of the Workshop Mobile Computer Systems and Applications*, IEEE CS Press, pp. 85-90, 1994.
- [2] S.Shafer, B.Brumitt, and B.Meyers, "The EasyLiving Intelligent Environment System," *CHI Workshop on Research Directions in Situated Computing*, 2000.
- [3] "Sensing the Subtleties of Everyday Life," *the research magazine of Georgia Tech*. (Research Horizons), 2000.
- [4] M.C.Mozer, "An intelligent environment must be adaptive," *IEEE Intelligent Systems and their Applications*, 14(2), pp. 11-13, 1999.
- [5] House_n Research Group, http://architecture.mit.edu/house_n/, 2005.
- [6] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications", *In Proceedings of the Workshop Mobile Computer Systems and Applications*, IEEE CS Press, pp. 85-90, 1994.
- [7] D. Salber, A.K. Dey and G.D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," *In Proceedings of the ACM Conf. on Human Factors and Computing Systems (CHI 1999)*, pp. 434-441, 1999.
- [8] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. "Advanced interaction in context," *In Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC 1999)*, Lecture Notes in Computer Science, 1707, pp. 89-101, 1999.
- [9] Tao Gu, Hung Keng Pung, Da Qing Zhang, "Toward an OSGi-Based Infrastructure for Context-Aware Applications," *IEEE Pervasive Computing*, Vol.03, No.4, pp. 66-74, 2004.
- [10] Jakob E. Bardram. "The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications," *In Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005)*, Lecture Notes in Computer Science, 3468, pp. 98-115, 2005.
- [11] Nabih Belhanafi Behloul, Chantal Taconet and Guy Bernard, "An Architecture for supporting Development and Execution of Context-Aware Component applications," *In IEEE International Conference on Pervasive Services, ICPS 2006*, pp. 26-29, 2006.
- [12] 홍동표, 신춘성, 오세진, 우운택, "사용자-콘텐츠-환경간의 이음매없는 상호작용을 위한 프레임워크," *SK텔레콤 Telecommunications Review*, 제 16권 제 4호, pp. 644-661, 2006.
- [13] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," *In Proceedings of PERVASIVE 2004*, LNCS 3001, pp. 1-17, 2004.
- [14] K. Van Laerhoven, A. Schmidt and H.-W. Gellersen. "Multi-Sensor Context-Aware

Clothing," *In Proc. of the 6th Int. Symposium on Wearable Computers, ISWC 2002*, IEEE Press, pp. 49-57, 2002.

- [15] Y.Oh, C.Shin, S.Jang and W.Woo, "ubi-UCAM 2.0: A Unified Context-aware Application Model for Ubiquitous Computing Environments," *The first Korea/Japan Joint Workshop on Ubiquitous Computing & Networking Systems (UbiCNS05)*, 2005.
- [16] 홍동표, 우운택, "맥락 인식 애플리케이션을 위한 사용자 중심의 맥락 모델," *한국정보처리학회 춘계학술발표대회 (KIPS)*, pp. 810-813, 2007.
- [17] S.Jang and W.Woo, "5W1H: Unified User-Centric Context," *The 7th International Conference on Ubiquitous Computing(UbiComp 05)*, 2005.



신 춘 성

- 2004년 숭실대학교 컴퓨터학부(공학사)
- 2006년 광주과학기술원 정보통신공학과(공학석사)
- 2006년~현재 광주과학기술원 정보기전공학부 박사과정
- 관심분야 : 유비쿼터스 컴퓨팅, HCI, 맥락인식 컴퓨팅, 스마트 홈, 모바일 컴퓨팅 등



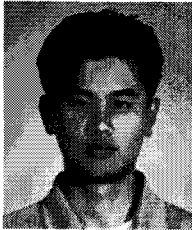
오 유 수

- 2002년 경북대학교 전자전기공학부(공학사)
- 2003년 광주과학기술원 정보통신공학과(공학석사)
- 2003년~현재 광주과학기술원 정보기전공학부 박사과정
- 관심분야 : 유비쿼터스 컴퓨팅, 맥락인식 컴퓨팅, HCI, 모바일/웨어러블 컴퓨팅 등



서 영 정

- 2001년 전남대학교 컴퓨터공학과(공학사)
- 2003년 광주과학기술원 정보통신공학과(공학석사)
- 2003년~현재 광주과학기술원 정보기전공학부 박사과정
- 관심분야 : HCI, 유비쿼터스 컴퓨팅, 맥락인식 컴퓨팅, Mixed Reality 등



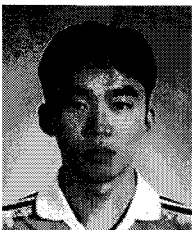
윤 호 석

- 2005년 숭실대학교 컴퓨터학부(공학사)
- 2007년 광주과학기술원 정보통신공학과 (공학석사)
- 2007년~현재 광주과학기술원 정보통신공학과 박사과정
- 관심분야 : 유비쿼터스 컴퓨팅, HCI, 모바일 컴퓨팅, 모바일 상호작용 기법 등



우 은 택

- 1989년 경북대학교 전자공학과(공학사)
- 1991년 포항공과대학교 전기전자공학과(공학석사)
- 1998년 University of Southern California, Electrical Engineering-System(공학박사)
- 1991년~1992년 삼성종합기술연구소(연구원)
- 1999년~2001년 ATR, Japan(초빙연구원)
- 2001년~현재 광주과학기술원 정보기전공학부(부교수)
- 2005년~현재 문화기술연구센터(센터장)
- 관심분야 : 3D 컴퓨터 비전, Attentive AR/mediated reality, HCI, Affective sensing, 맥락인식 컴퓨팅, 유비쿼터스 컴퓨팅 등



한 중 현

- 2005년 아주대학교 정보 및 컴퓨터공학부(공학사)
- 2007년 광주과학기술원 정보통신공학과(공학석사)
- 2007년~현재 광주과학기술원 정보통신공학과 박사과정
- 관심분야 : 유비쿼터스 컴퓨팅, 커뮤니티 컴퓨팅, Social-Awareness 등