

AIM: 유비쿼터스 HCI 환경을 위한 에이전트 기반 지능형 미들웨어 설계 및 구현

장 현 수^{*} · 김 연 우^{**} · 최 정 환^{***} · 강 동 현^{**} · 송 창 환^{**} · 엄 영 익^{****}

요 약

유비쿼터스 시대의 도래와 함께, 사용자 중심 서비스를 위해 HCI 요소를 최대한 활용하는 HCI 미들웨어에 대한 중요성이 점점 증가하고 있다. 이에 따라 사용자 중심 서비스를 제공하기 위한 HCI 미들웨어에 대한 여러 연구가 진행되었다. 하지만 기존의 연구들은 사용자 중심 서비스 제공에 반드시 필요한 HCI 기능들에 대한 종합적 지원이 부족하다. 본 논문에서는 사용자 중심 서비스를 위한 에이전트 기반 지능형 HCI 미들웨어인 AIM(Agent-based Intelligent Middleware)을 제안한다. 먼저 기존의 여러 HCI 미들웨어들을 분석하여 HCI 미들웨어의 요구조건을 도출하고, 상황 정보 관리, 사용자 행동 패턴 추론, 동적 에이전트 생성 등을 통해 여러 HCI 기능들을 종합적으로 제공할 수 있는 AIM을 설계한다. 또한 스마트 공간에서 사용자 모달리티 기반 서비스 시나리오를 소개하고, 프로토타입 구현을 통해 소개한 미들웨어의 이점을 보인다.

키워드: 휴먼-컴퓨터 인터랙션, 미들웨어, 에이전트, 유비쿼터스 컴퓨팅

AIM: Design and Implementation of Agent-based Intelligent Middleware for Ubiquitous HCI Environments

Hyunsu Jang^{*} · Younwoo Kim^{**} · Jung Hwan Choi^{***} · Dong Hyun Kang^{**}
· Changhwan Song^{**} · Young Ik Eom^{****}

ABSTRACT

With the emergence of ubiquitous computing era, it has become increasingly important for a middleware which takes full advantage of HCI factors to support user-centric services. Many kinds of studies on HCI-friendly middleware for supporting user-centric services have been performed. However, previous studies have problems in supporting HCI factors, which are needed for user-centric services. In this paper, we present an agent-based intelligent middleware, which is called AIM, that provides user-centric services in ubiquitous HCI environments. We describe the middleware requirements for user-centric services by analyzing various HCI-friendly middleware and design AIM middleware which effectively supports various HCI factors such as context information management, pattern inference of user's behavior, and dynamic agent generation, etc. We introduce service scenarios based on the user's modalities in smart spaces. Finally, prototype implementation is illustrated as a manifestation of the benefits of the introduced infrastructure.

Keywords: HCI, Middleware, Agent, Ubiquitous Computing

1. 서 론

가까운 미래에는 더욱 풍요로운 삶을 인간에게 제공하기

위해 손쉬운 서비스 접근 및 자연스러운 상호작용이 가능한 다양한 컴퓨팅 디바이스들이 유비쿼터스 컴퓨팅 환경 곳곳에 존재하게 될 것이다. 이에 따라 최근 많은 연구자 및 기관들은 상황인지(context-awareness), 적응성(adaptation), 재구성(reconfiguration) 등과 같은 개념을 가진 여러 유비쿼터스 컴퓨팅 미들웨어들을 개발하고 있다.

HCI는 유비쿼터스 컴퓨팅 환경에서 인간과 장치 사이의 편리한 상호작용을 연구하는 학문분야이며[1][2], 상호작용 컴퓨팅 시스템을 설계, 평가 및 구현 하는 것 또한 포함한다. 이러한 HCI에 대한 연구는 다양한 장비가 곳곳에 존재

* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(ITA-2008-(C1090-0801-0046)).

[†] 준 회 원: 성균관대학교 전자전기컴퓨터공학과 박사과정

^{**} 준 회 원: 성균관대학교 전자전기컴퓨터공학과 석사과정

^{***} 준 회 원: 성균관대학교 휴대통신학과 석사과정

^{****} 종신회원: 성균관대학교 정보통신공학부 교수

논문접수: 2008년 10월 16일

수정일: 1차 2009년 1월 14일

심사완료: 2009년 1월 14일

하는 유비쿼터스 환경에서 사용자에게 자연스럽고 편리한 서비스 제공을 가능케 한다. 이에 따라 유비쿼터스 컴퓨팅 미들웨어는 HCI 지원 컴포넌트를 포함하거나, 다른 독립적인 HCI 미들웨어와의 상호작용을 제공하는 등 HCI 요소에 대한 개발 및 개선이 반드시 필요하다.

현재 사람의 모달리티(음성, 제스처 등) 인식에 기반을 둔 홈 리빙 서비스와 생체인식을 통한 보안 서비스 등의 HCI 서비스가 홈 네트워크 분야에서 부분적으로 개발 및 서비스되고 있다. 그러나 이것들은 HCI 개념을 도입한 초기 단계에 지나지 않으며, 유비쿼터스 환경은 더욱 적극적인 HCI 기능의 지원을 필요로 한다..

본 논문에서는 에이전트 기반 지능형 HCI 미들웨어인 AIM(Agent-based Intelligent Middleware)을 제안한다. 제안된 AIM은 XML 기술에 기반하여 사용자, 장치, 이벤트와 상황 프로파일 등을 관리한다. 그리고 이러한 프로파일로부터 정보를 추출 및 조합하여, 사용자에게 적응적 UI를 제공한다. 그리고 사용자의 모달리티 정보와 장치로부터 수집한 여러 상황정보를 분석 및 통합하여 사용자와 장치간의 자연스런 상호작용을 지원한다. 또한 다양한 기기종 장치, 미들웨어, 그리고 응용 사이의 통신적 상호운용성과 의미론적 상호운용성을 제공한다. 마지막으로 시스템의 기본 레이어로서 사용자 중심 서비스를 위한 지능형 에이전트를 지원하는 이동 에이전트 시스템을 제공한다.

본 논문의 각 장은 다음과 같은 내용으로 구성되어 있다. 2장에서는 HCI 요소를 접목한 기존 미들웨어들을 살펴본다. 3장에서는 HCI 미들웨어에 대한 요구사항을 도출하고, 이를 만족하는 AIM 시스템 아키텍처를 설명한다. 또한 이를 구성하는 여러 HCI 지원 컴포넌트와 에이전트 플랫폼에 대해 설명한다. 4장에서는 AIM의 검증을 위한 구현 및 평가 결과를 보인다. 5장에서 본 논문을 마무리하고 향후 연구 과제와 쟁점을 소개한다.

2. 유비쿼터스 환경을 지원하는 사용자 중심 미들웨어

유비쿼터스 컴퓨팅 분야에서 사용자 중심 서비스는 최근의 중요한 연구 주제이다. 본 절에서는 사용자 중심 서비스 지원을 위해 HCI 요소를 접목한 여러 미들웨어 연구들에 대해 살펴본다.

일본 와세다 대학에서는 2001년에 사용자 선호도에 따라 선택된 입출력 장치를 통해, 어디서든 단일한 방법으로 네트워크 정보 가전을 제어할 수 있는 썬 클라이언트(Thin-Client) 기반 미들웨어를 제안하였다[4]. 이 미들웨어는 사용자의 선호도와 현재 상황, 디바이스를 고려한 인터페이스 적응성을 지원하고 있지만, 멀티 모달리티를 지원하지 못하였다.

Célestin Sedogbo & Human Interaction Technologies Lab은 2003년에 응용 및 인터페이스에 독립적인 적응적 상호작용 서비스를 제공하기 위해서 Human-System Interaction Container(HIC) 및 에이전트 기반 상호작용 미들웨어를 제안하였다[5]. 이 미들웨어는 적응적 상호작용의 구현을 위해

소프트웨어 컴포넌트의 캡슐화를 통한 응용, 상호작용, 프리젠테이션 로직 간의 결합성을 지원한다. 이를 통해, 응용의 변경 없이도 동적인 적용이 가능한 HIC 개념을 제시하였다. 또한 사용자의 제어 없이도 에이전트 협업을 통해 다양한 목적을 지닌 응용들 간의 동적 환경 적응성을 제공할 수 있다. 이를 바탕으로 이 미들웨어는 멀티모달 상호작용, 개인화된 사용자 인터페이스 적용 및 이동 에이전트 기반 서비스를 지원한다.

핀란드 Oulu 대학의 Pertti Repo와 Jukka Riekkii는 2004년에 Context-Aware Pervasive Network(CAPNET) 환경에 존재하는 다양한 기기종 디바이스 및 사용자 컨텍스트에 적응적인 사용자 인터페이스를 자동으로 생성해주는 CAPNET 미들웨어 아키텍처를 제안하였다[6]. 이 미들웨어는 핵심 컴포넌트인 Capnet User Interface Component는 XML 기반 사용자 및 디바이스 정보를 이용하여 적응적이고 플랫폼 독립적인 사용자 인터페이스를 제공한다.

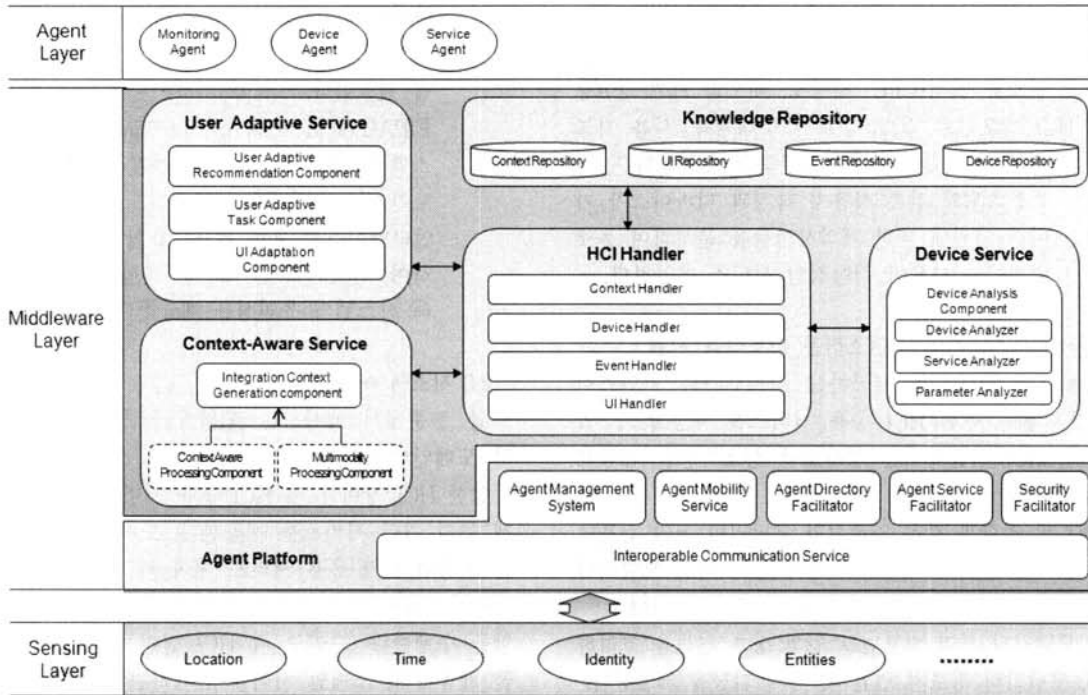
스위스 Fribourg 대학은 2005년에 Context-Aware Middleware for Multimodal Dialogue Applications(CAMA) 프로젝트에서 음성, 제스처, 표정 등을 사용해 시스템과 자연스러운 상호작용을 할 수 있는, 멀티모달 다이얼로그 어플리케이션을 위한 에이전트 기반 상황인지 미들웨어인 CAMA 미들웨어를 제안하였다[7]. CAMA 미들웨어는 자원, 상황, 계획, 컨텍스트 이력, 데이터 등의 개념을 설명하는 온톨로지 모델을 통해, 의미론적 상호운용성(semantic interoperability)과 상황인지를 지원한다. 또한, 컨텍스트 이력 관리자 에이전트(Context History Manager agent-CHMa)를 통해 컨텍스트에 대한 처리 결과를 관리함으로써, 컨텍스트 추적(context tracing) 기능을 제공한다.

본 논문에서 제시하는 AIM은 위 HCI 미들웨어에 대한 분석을 통해 설계된 아키텍처이다. AIM은 기본 컴포넌트와 확장 컴포넌트의 결합 구조를 통해, 사용자 및 디바이스에 대한 적응적인 멀티 모달 사용자 인터페이스의 제공이 가능하며, 기기종의 디바이스, 응용, 또는 시스템 사이의 의미론적 상호운용성을 제공할 수 있을 것으로 기대된다.

3. AIM (Agent-based Intelligent Middleware)

본 논문에서 제시하는 추상 HCI 미들웨어 참조 모델은 FIPA 에이전트 관리 시스템[8]을 기반으로 HCI 응용 서비스에 필요한 기능들을 제공함으로써, 효율적이고 편리한 사용자 중심의 자연스런 상호작용을 지원한다. 다음 (그림 1)은 본 논문에서 제안하는 추상 HCI 미들웨어 참조 모델을 보인다.

(그림 1)에서 보이는 바와 같이, 본 참조 모델은 센싱 계층, 미들웨어 계층, 에이전트 계층 총 3개의 계층 구조를 가진다. 본 논문에서 제시하는 에이전트 기반 지능형 미들웨어 AIM은 HCI 미들웨어 참조 모델의 미들웨어 계층과 에이전트 계층으로 구성된다. 그리고 센싱 계층은 AIM과의 연결을 통해 센싱 정보를 공급한다. 위 참조 모델에서는 언



(그림 1) 추상화 에이전트 기반 HCI 미들웨어 참조 모델

급하지 않았지만, 본 미들웨어는 운영체제와 가상머신 Java 2 Micro Edition(J2ME)에 기반을 둔다. 본 참조 모델에서는 J2ME를 지원하지 않는 컴퓨팅 디바이스를 제외한 모든 디바이스에서 사용될 수 있다.

미들웨어 계층은 에이전트 시스템과 HCI 컴포넌트로 구성된다. 첫째, 에이전트 시스템은 이동성, 자율성, 지능성, 사회성을 갖는 이동 에이전트 기술을 제공한다. 특히, 에이전트 시스템의 ICS(Interoperable Communication Service)는 에이전트 간 또는 HCI 지원 컴포넌트 간의 상호운용적인 메시지 통신을 지원한다. 이러한 ICS의 통신 기능을 통하여, 이기종 플랫폼과의 상호운용성 문제를 최소화 한다. 둘째, HCI 컴포넌트들은 CAS(Context-Aware Service), UAS(User Adaptive Service), DS(Device Service), HCI Handler, KR(Knowledge Repository)로 구성된다. CAS는 센싱 계층과의 통신을 통해 여러 센싱 데이터를 받아 사용자 중심 서비스를 제공하기 위한 사용자, 장치, 이벤트와 상황 정보를 축적한다. UAS는 사용자 인터페이스, 시스템 상황과 사용자의 요구에 따라 적응적인 서비스 제공을 위해서, 축적된 센싱 정보를 기반으로 여러 데이터마이닝 기법들에 근거한 지능형 서비스를 제공한다. DS는 물리적으로 독립되어 유비쿼터스 서비스를 제공하는 여러 HCI 장비를 제어하기 위해, 장비의 등록, 삭제, 분석 등 관리를 수행한다. HCI Handler는 미들웨어 계층 내에서 이루어지는 모든 메시지의 전달을 관리하여 서비스 흐름을 제어한다. 마지막으로 KR은 다른 HCI 지원 컴포넌트들이 사용하는 여러 정보들을 관리한다.

상단의 에이전트 계층은 사용자와 직접 상호작용하는 에이전트로 구성된다. Monitoring Agent는 각 플랫폼 및 장비의 상태와 에이전트들의 정상 동작 여부를 점검한다. Device Agent는 이동 에이전트 플랫폼의 지원을 바탕으로 HCI 장

치의 논리적 통신을 수행하는 인터페이스 역할을 담당한다. 마지막으로 Service Agent는 에이전트 시스템에서 제공하는 에이전트의 이동성, 지능성을 활용하여 HCI 컴포넌트의 지원과 Device Agent와의 통신을 통해 사용자의 요구에 적합한 서비스를 제공한다.

3.1 사용자 중심 서비스를 위한 미들웨어 요구조건

본 섹션에서는 사용자 중심 서비스를 하기 위해서 HCI 미들웨어의 정의와 요구조건을 설명한다. HCI 미들웨어는 사용자를 위한 편리하고 자연스런 상호작용 기능을 지원하는 미들웨어이다. HCI 미들웨어는 인간의 다양한 모달리티 사용에 필요한 여러 기술들을 지원하기 위해, HCI 서비스 구현에 사용되는 개발자용 API와 효율적이며 편리한 인터페이스 기능 등을 제공한다. 이를 통해 HCI 미들웨어는 사용자, 개발자, 관리자, HCI 장비, HCI 서비스들 간의 중간 조정자 역할을 수행한다. 이러한 HCI 미들웨어는 일반적인 미들웨어 요구사항과 더불어 다음과 같은 세 가지 특징적인 요구사항을 만족해야 한다. 본 논문에서는 다음과 같은 방법을 통하여 요구사항에 접근한다.

● 사용자 중심의 서비스 지원

- HCI 미들웨어는 사용자들에게 투명하고 최적화된 서비스를 제공하기 위해서 사용자 특성 및 HCI 장비의 특징을 고려하여 적응성(adaptation) 기능을 제공해야 한다.
- 조적이고 확장 가능한 XML 기반의 사용자 프로파일을 통해, 사용자 관련 정보를 정적(계정, 성별, 주소 등)/주기적(역할, 권한, 장애 정보 등)/동적(위치,

시간, 상황 등) 정보로 분류하고 제공한다.

- XML 기반의 HCI-장비 프로파일을 통해 HCI 장비 관련 정보를 정적(CPU, 해상도, 메모리 크기 등)/동적(HCI 컴포넌트 정보, 네트워크 대역폭, 사용 가능한 메모리 크기 등) 정보로 분류하고 제공한다.
- 사용자 선호도와 같은 사용자 특징과 디바이스가 가지는 여러 자원에 대한 제약사항들을 분석하여 플랫폼 독립적이고 사용자 적응적인 서비스 제공한다.

● 사용자와 시스템간의 자연스러운 상호작용 지원

- 사용자와 서비스를 제공하는 시스템간의 자연스런 상호작용을 통해 HCI 응용 서비스를 제공할 수 있도록, HCI 미들웨어는 멀티모달 상호작용을 제공해야 한다.
- 시각 및 청각과 같은 기본적인 모달리티 처리 컴포넌트를 기반으로, 후각, 미각, 촉각 등의 다른 모달리티를 확장 지원할 수 있는 인터페이스를 제공한다.
- 각 모달리티 처리 컴포넌트 간의 상호 연동을 고려하여, 모달리티 정보의 변환 및 중재 기능을 제공한다.
- 사용자가 가질 수 있는 장애여부, 위치 등의 특징을 고려하여 HCI 응용 서비스를 효과적으로 전달할 수 있는 메커니즘을 제공한다.

● 시스템간 상호운용성 지원

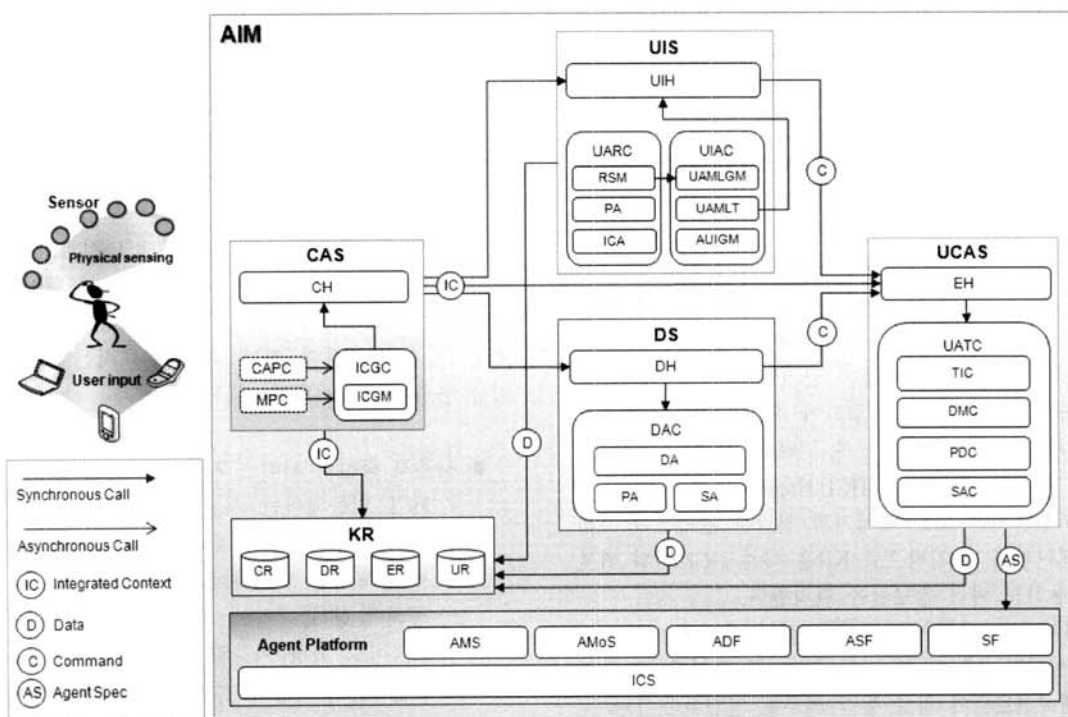
- 여러 기기종 시스템 간의 정보 교환과 서비스 호출을 위하여, 다양한 에이전트 관련 언어 지원을 통해 통신 상호운용성과 의미론적 상호운용성을 제공한다.
- ACL(Agent Communication Language), KQML(Knowledge

Query and Manipulation Language) 등 다양한 에이전트 관련 언어를 사용하기 위한 변환기(Parser)등을 제공한다.

- FIPA(Foundations of Intelligent Physical Agents) 스펙을 따르는 에이전트 플랫폼을 제공하여 에이전트간 상호운용성을 제공한다.
- 인터페이스를 통한 컴포넌트의 접근 및 구현을 지향하여, 컴포넌트 간 느슨한 결합(loose coupling)을 통해 시스템의 유연성을 제공한다.

3.2 시스템 아키텍처

본 논문에서 제시하는 AIM의 시스템 인프라스트럭처는 각 도메인별로 도메인 서버를 두어, 도메인 내에 존재하는 다양한 HCI 장비들 및 HCI 응용 서비스들에 대한 등록, 해제 등의 관리 서비스를 제공한다. 도메인 내의 도메인 서버는 최상위 HCI 응용 서비스 에이전트를 제공하여 사용자가 요구하는 HCI 응용 서비스에 대한 접근점(access point) 역할을 수행한다. 그리고 각 HCI 장비는 자신의 HCI 서비스를 제공하며, 도메인 서버에 존재하는 최상위 HCI 응용 서비스 에이전트에 의해 제어되는 구조를 가지고 있다. 그리고 특정한 HCI 응용 서비스의 수행을 위하여, 서비스 에이전트에 의한 HCI 컴퓨팅 디바이스 간 상호연동을 제공한다. (그림 2)는 본 논문에서 제안하는 AIM을 통하여 HCI 응용 서비스를 구축하기 위한 시스템 인프라스트럭처를 보인다. 이 시스템 인프라스트럭처는 HCI 지원 컴포넌트를 통하여 사용자 중심 서비스를 중점적으로 제공한다. HCI 지원 컴포넌트는 CAS(Context-Aware Service), UIS(User Interface Service), UCAS(User-Centric Adaptive Service)와 DS(Device



(그림 2) 미들웨어 동작 흐름도

Service)로 구성된다. 각각의 HCI 지원 컴포넌트에 대해 아래와 같이 간략히 설명하고 세부적인 서브 컴포넌트의 내용은 3.3절에서 자세히 설명한다.

- CAS는 상황 정보 및 멀티모달리티 입력을 위한 CAPC (Context Aware Processing Component)와 MPC (Multimodality Processing Component), 컨텍스트를 생성하는 ICG(Integration Context Generation Component)와 크게 메시지 송수신을 담당하는 핸들러인 CH(Context Handler)로 구성된다.
- UCAS는 관련 메시지 전달을 담당하는 EH(Event Handler)와 사용자 적응적 서비스를 위한 추론 및 Service Agent 생성을 담당하는 UATC(User Adaptive Task Component)로 구성된다.
- UI Service는 크게 메시지 송수신을 담당하는 핸들러인 UIH(UI Handler)와 적응형 UI를 제공하는 컴포넌트인 UIAC(UI Adaptation Component), 그리고 사용자 적응형 추천 UI를 제공하는 UARC(User Adaptive Recommendation Component)로 구성된다.
- DS는 장비 서비스와 관련된 모든 메시지를 처리하는 DH(Device Handler)와 장비 사용 내역과 상황정보를 바탕으로 장비를 분석하여 서비스 요청을 구체화하는 DAC(Device Analysis Component)로 구성된다.

또한, 제안하는 AIM은 다음 과정을 통해 사용자 중심 서비스를 제공한다.

- 센싱 정보 수집 : 도메인 내의 각 센서들은 온도, 밝기 등 여러 상황정보와 시각적 또는 청각적으로 전해지는 사용자의 행동 정보를 수집한다.
- CAS의 IC 생성 : CAS의 CAPC, MPC는 각각 여러 센서로부터 상황정보와 사용자의 행위정보를 수집한다. 이를 바탕으로 IC(Integrated Context)를 생성한다.
- UCAS의 사용자 행동 추론과 서비스 결정 : UCAS는 생성된 IC를 추론한다. 저장된 과거의 상황정보 및 사용자의 행동 정보를 바탕으로 사용자의 행동을 모델링한 후, 이를 바탕으로 사용자의 요구를 추천하여 사용자에게 추천 서비스와 선응적 서비스를 제공하기 위한 Service Agent를 생성한다.
- UIS와 DS를 이용한 Service Agent의 서비스 제공 : 생성된 Service Agent는 UIS와 DS를 사용하여 사용자 적응적 UI를 제공하고 장비를 제어하여 사용자 중심 서비스를 제공한다.

각각의 컴포넌트들은 위와 같은 흐름의 동작을 통해 사용자 중심 서비스를 제공한다. 또한 에이전트 플랫폼은 동적 에이전트 생성과 통신 상호운용성 지원을 위한 코어 모듈 기능을 제공한다. 그리고 각 컴포넌트들의 정보를 KR(KnowledgeRepository)에 보관함으로써 사용자 중심 서비스를 제공한다.

3.3 HCI 지원 컴포넌트

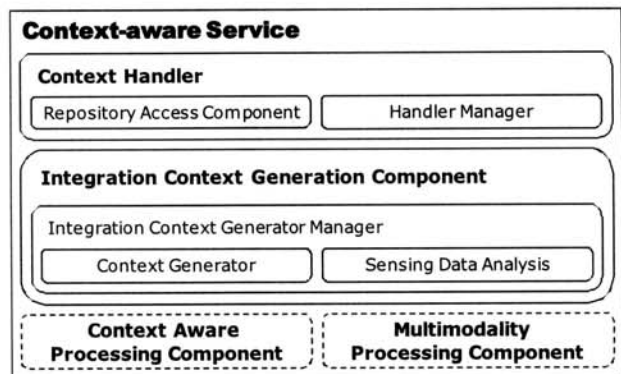
3.3.1 CAS (Context-Aware Service)

CAS는 사용자의 명령 등이 기록된 사용자의 행위 정보와 사용자 주위의 상황정보를 기반으로 다른 어플리케이션들을 위한 Integrated Context(IC) 생성 및 관리 서비스를 제공한다. 즉, 단말의 UI 및 사용자의 멀티 모달리티 처리로부터 입력된 사용자의 명령과 온도센서, 광 센서 등과 같은 사용자의 주변 환경 정보의 주기적 인식을 통해 IC를 생성 및 전송한다.

CAS에서는 환경 정보를 인식하는 센서들 간의 상이한 프로토콜 처리를 위해 프로토콜 에이전트를 사용한다. 또한 중복된 센서 정보로 인한 시스템 부하를 줄이기 위하여, 개미 군집 시스템 알고리즘을 이용하여 컨텍스트 생성 여부를 판단한다. 또한 IC를 기반으로 사용하는 어플리케이션 및 장치들을 위하여, IC를 장치 및 서비스에 독립적인 XML 기반으로 작성한다.

CAS는 상황 정보 및 멀티모달리티 입력을 위한 CAPC와 MPC, 컨텍스트를 생성하는 ICG와 크게 메시지 송수신을 담당하는 핸들러인 CH로 구성된다. (그림 3)은 CAS의 내부 구조도를 보인다.

- CAPC(Context-Aware Processing Component)와 MPC (Multi-modality Processing Component) : CAPC는 여러 센서로부터 수신한 상황 정보를 입력받아 처리한다. MPC는 UI를 비롯하여 음성, 제스처, 터치 등 사용자의 명령인식에 대한 멀티 모달리티를 지원한다. CAPC는 미들웨어의 주기적 요청에 의해 상황 정보를 수집하며, MPC는 각 입력장치로부터 사용자의 명령을 인식하였을 때 명령을 수집한다. 이렇게 수집된 상황정보와 명령은 ICG로 전달된다. 또한 CAPC와 MPC는 인터페이스로 제공되어 외부 라이브러리에 의해 구현된다.
- ICG(Integrated Context Generation Component) : ICG에서는 상황에 맞는 적절한 서비스를 사용자에게 제공하기 위해서 사용자의 위치, 사용자 주변 환경, 사용 가능한 자원 등의 정보를 XML기반의 컨텍스트로 생성하는 역할을 담당한다. 또한 ICG는 개미 군집 시스템 알고리즘을 이용하여 컨텍스트 생성 여부를 판단하고, 저장소를 이용하여 중복된 컨텍스트



(그림 3) 상황인식 서비스 컴포넌트

적재를 방지한다. 이렇게 수신한 각각의 정보를 분류하여 트리 구조로 그룹화한 후, 사용자에게 적응적인 서비스를 제공하기 위하여 사용자의 환경정보 및 명령을 장치에 독립적인 XML기반의 IC를 생성한다. ICG에서 생성된 IC는 CH로 전송된다.

- CH(Context Handler) : CH는 IC와 관련된 모든 외부와의 입출력을 담당한다. CH의 세부 컴포넌트인 HM(Handler Manager)은 ICG로부터 생성된 IC들을 받아들이고, IC를 기반으로 동작하는 응용 어플리케이션을 위하여 IC를 EH(Event Handler)로 전송한다. 또한 생성된 IC는 RAC(Repository Access Component)을 통하여 KR에 저장된다.

3.3.2 UCAS (User-Centric Adaptive Service)

UCAS는 사용자 중심의 적응성을 제공하는 서비스이다. 상황 정보와 사용자의 과거 행위 정보를 바탕으로 사용자의 요구를 추론한 후, 사용자에게 적응적 서비스와 선용적 서비스를 제공한다. 적응적 서비스는 현재 상황 정보를 바탕으로 생성한 사용자의 평소 생활 패턴을 기반으로, 사용자에게 필요한 서비스를 선택하여 제공하는 것이다. 또한 선용적 서비스는 사용자가 요구할 서비스를 예측하여, 서비스를 미리 제공하는 것이다.

이러한 서비스를 제공하기위해 UCAS는 CAS로부터 상황 정보와 사용자의 행위 정보가 기록된 IC를 받는다. 상황 정보와 사용자의 행위정보는 데이터 마이닝 기법을 이용하여 사용자가 어떤 상황에서 어떤 장치에 어떤 명령을 수행하는지에 대한 패턴을 추론하는 데에 사용된다. UCAS는 이 추론 결과를 바탕으로 Service Agent를 생성하여 사용자에게 서비스를 제공한다.

UCAS는 관련 메시지 전달을 담당하는 EH와 사용자 적응적 서비스를 위한 추론 및 Service Agent 생성을 담당하는 UATC로 구성된다. (그림 4)는 UCAS의 내부 구조도를 보인다.

- EH (Event Handler) : UATC가 요구하는 IC를 CAS로부터 수신하여 전달하거나, UATC로부터 받은 추론 결과를 KR에 저장하는 등, 데이터를 전달해주는 역할을 한다.
- UATC (User Adaptive Task Component) : UATC는 EH로부터 받은 IC를 이용해 사용자의 행동을 분석하여 사용자의 요구를 추론하고, 사용자의 요구를 충족시킬 수 있도



(그림 4) 사용자 중심 적응적인 서비스 컴포넌트

록 서비스를 구성하여 이를 수행할 수 있는 Service Agent를 생성한다. TM(Task Manager)는 EH로부터 받은 IC를 추적하여 사용자의 행동 패턴을 모델링한다. 이때 PDC(Pattern Discovery Component)로 사용자의 누적된 행동 정보에서 연관성 있는 행동들을 추출하고, DMC(Decision Making Component)의 데이터 마이닝 기술로 사용자의 행동 규칙을 찾아낸다. 이렇게 모델링이 완료되면 새로운 IC를 전달받았을 때 사용자의 행동 패턴과 IC를 바탕으로 사용자에게 필요한 서비스를 선택한다. SAC(Service Agent Composer)는 이러한 과정을 통해 선택된 서비스 목록들을 수신하여, 이 서비스들을 수행할 수 있는 태스크들을 미리 저장된 KR에서 가져와 Service Agent 생성을 위한 AS(Agent Specification)을 만들어 에이전트 플랫폼에게 전달한다. 에이전트 플랫폼은 수신한 AS으로 Service Agent를 생성한다. 생성된 Service Agent는 UIS와 DS를 제어하여 사용자 중심 서비스를 제공한다.

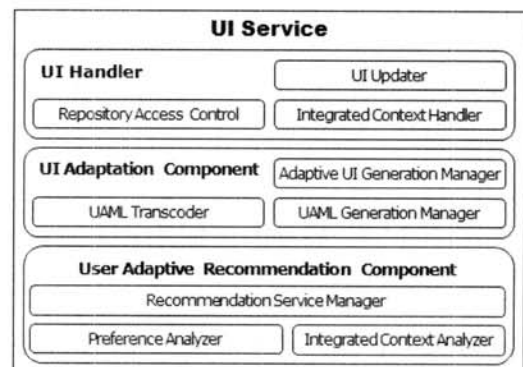
3.3.3 UIS (User Interface Service)

UIS는 주변의 상황 정보 및 HCI 디바이스들의 제약 사항을 분석하여, 사용자와 시스템간의 자연스런 상호작용이 가능한 적응적인 UI를 제공한다. 적응형 UI를 생성 및 표시하기 위해 AIM에 적합한 XML기반의 사용자 적응적인 UI 기술 언어인 UAML(User Adaptive Markup Language)을 이용한다. UIS는 기기종 디바이스 간 이질성을 UAML이란 하나의 범용적인 언어 형태로 구축함으로써, 장치 독립성과 원활한 유지보수를 가능하게 한다.

UIS는 사용자에게 대한 선호도 정보와 장치들에 대한 세부 내용이 포함된 프로파일들을 참조하여, 해당 장치에 맞는 UAML을 생성한다. 이 정보들은 추천 리스트와 사용자 선호도에 기반을 두어 적응형 UI를 생성한다. 생성된 적응형 UI는 사용자 장치에 독립적인 형태로 변형된 후 사용자 개인에게 맞는 맞춤형 UI로 제공된다.

UI Service는 크게 메시지 송수신을 담당하는 핸들러인 UIH와 적응형 UI를 제공하는 컴포넌트인 UIAC(UI Adaptation Component), 그리고 사용자 적응형 추천 UI를 제공하는 UARC(User Adaptive Recommendation Component)로 구성된다. (그림 5)는 UI 서비스 컴포넌트의 내부 구조도를 보인다.

- UIH (UI Handler) : UIH는 UI와 관련된 외부와의 모



(그림 5) UI 서비스 컴포넌트

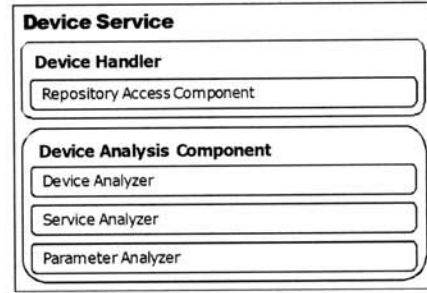
든 입출력을 담당한다. CAS로부터 IC를 받아 적응형 추천 서비스를 제공하는 UARC로 보내며, KR로의 접근을 통해 장치 혹은 사용자 등에 대한 데이터를 가져 오거나 입력을 수행한다.

- UIAC (UI Adaptation Component) : UIAC에서는 사용자 프로파일 및 장비 제약 속성을 고려하여 적응형 UI를 생성한다. 우선 UAMLGM(UAML Generation Manager)는 디바이스 프로파일로부터 디바이스 정보를 획득하여 각 디바이스에 적합한 UAML을 생성한다. 디바이스 프로파일 내에는 각 장치들에 대한 상세정보가 포함되어 있고, UAMLGM에서는 이들 정보를 바탕으로 해당 장치를 조작할 수 있는 UI를 UAML 형태로 작성한다. 이는 AUGM(Adaptive UI Generation Manager)로 보내져 UARC의 RSM(Recommendation Service Manager)으로부터 전송 받은 사용자 프로파일 기반의 추천 스타일 목록과 함께 분석되어, 적응형 UI에 대한 UAML을 새롭게 생성하게 된다. 이후 UAMLT(UAML Transcoder)은 적응형 UAML를 각 장치에 적합한 UI로 변환하여 전송한다.
- UARC (User Adaptive Recommendation Component) : UARC는 사용자의 정적 프로파일 및 동적 프로파일을 분석하여, 사용자 적응적 UI 스타일 선택 및 UI 스타일의 추천 리스트 생성을 담당하는 컴포넌트이다. 전체적인 흐름을 관리하는 RSM(Recommendation Service Manager)와 상황 정보를 분석해서 선호도를 측정하는 PA(Preference Analyzer)로 구성된다. 상황 인지 서비스에 의해 IC가 입력되면 ICA(Integrated Context Analyzer)가 IC를 분석한다. 분석을 통해 장치별 스타일 선호도를 계산한다. 계산된 선호도의 우선 순위에 따라 RSM에서는 UI 스타일의 추천 리스트를 생성하고, 적응형 UI 생성을 위해 이를 UIAC의 AUGM으로 전송한다.

3.3.4 DS (Device Service)

장치 서비스는 AIM에 연결된 HCI 장비들을 관리한다. 장치 서비스는 사용자를 위한 장비 제어, 장비 추천, 사용 내역 저장 기능들을 제공한다. 이 기능들을 제공하기 위해, 탐색, 등록, 추상화, 메시지 전달, 추론, 자료 갱신 등의 업무를 수행한다.

장비 서비스는 장비 서비스와 관련된 모든 메시지를 처리하는 DH(Device Handler)와 장비 사용 내역과 상황정보를 바탕으로 장비를 분석하여 서비스 요청을 구체화하는 DAC(Device Analysis Component)로 구성된다. 또한 메인 서버 외부의 각 장비에 상주하는 Device Agent, Monitoring Agent 두 에이전트와 밀접한 통신을 수행한다. Device Agent는 물리 장비와의 연결을 통해 실제 서비스를 제공한다. Monitoring Agent는 Device Agent의 서비스 수행과 정상 동작 여부를 감시한다. (그림 6)은 장비 서비스 컴포넌트의 내부 구조도를 보인다.



(그림 6) 장비 서비스 컴포넌트

- DH (Device Handler) : DH는 외부로부터 서비스 요청을 받아 결과를 전달하고, 서비스 내부에서 이루어지는 업무 수행을 관리한다. 장비서비스 외부의 Service Agent로부터 서비스 요청을 받아들여 DH의 작업 목록에 수신한 서비스 요청을 추가한다. DH는 자신의 작업 목록을 관리함으로써 장비서비스의 정상 작동 여부를 관리하고, 만약 내부에서 처리중인 작업이 저장소의 자료를 요구할 경우, KR과의 통신을 통해 자료를 전달한다.
- DAC (Device Analysis Component) : DAC는 구체적인 외부의 서비스 요청을 분석하여 Device Agent의 서비스 제공을 보다 쉽게 한다. DAC 먼저 장비 사용 내역과 현 서비스 요청을 비교한다. 만약, 현 서비스 요청의 목적 장비, 목적 서비스, 주요 인자 등 주요 인자가 일치하는 과거의 서비스 요청이 존재할 경우, 과거의 자료를 이용해 목적 장비와 서비스를 선택한다. 만약 과거의 서비스 요청이 존재하지 않을 경우, 내용기반 필터링을 사용한다. 미리 입력된 정보들, 장비에 관한 정보, 서비스에 관한 정보, 인자에 관한 정보를 각각 이용하여 장비별, 서비스별, 인자별로 주요 인자를 선택한다. DAC의 내부는 DA(Device Analyzer), SA(Service Analyser), PA(Parameter Analyser)가 각각의 필터링을 담당한다.
- Device Agent : Device Agent는 AIM과 물리 장비의 중간에 인터페이스 역할을 수행한다. AIM의 DH로부터 서비스 콜을 수신하면, 이 메시지를 바탕으로 물리 장비의 서비스를 구동시킨다. 이후 서비스가 종료되면 결과 값을 다시 AIM의 DH에게 넘긴다.
- Monitoring Agent : Monitoring Agent는 Device Agent를 감시하기 위한 이동 에이전트이다. 네트워크 내의 플랫폼 사이를 이주하며, Device Agent의 라이프사이클과 서비스의 정상 수행 여부를 확인한다. 만약 비정상 수행 또는 오류가 발생할 경우, 이러한 오류 사항들을 DH에게 보고한다.

3.4 에이전트 플랫폼 (Agent Platform)

유비쿼터스 환경에서 사용자는 셀 수 없을 만큼 많은 컴퓨팅 장비들로부터 서비스 받는다. 이에 따른 장비 사용의 어려움과 서비스의 복잡함을 해결하기 위하여, 사용자를 대

신해 작업을 처리하여 사용자의 개입을 최소화할 수 있는 에이전트 기술이 필요하다.

본 논문에서는 사용자 중심 서비스를 제공하는 미들웨어를 지원하기 위해, 본 연구실에서 개발한 LMAP(Lightweight Mobile Agent Platform)[9]을 개량하여 HCI 서비스에 맞는 이동 에이전트 플랫폼을 개발하였다. 본 에이전트 플랫폼은 LMAP과 같이 FIPA(Foundations of Intelligent Physical Agents) 스펙을 따른다. HCI 서비스에서는 사용자의 요청에 대한 가능한 빠른 응답이 요구되며, 사용자와의 상호작용하는 장비와 실질적인 서비스를 제공하는 장비 간에 집중적인 메시지 통신이 이루어지는 특성을 가지고 있다. 이러한 HCI 서비스의 특성을 고려하여, 기존 에이전트 플랫폼의 메시지 통신 기능에 인증/목적주소탐색/우선순위 스케줄링들을 개량하여 메시지 통신의 속도를 높이고 네트워크 부하를 줄였다. 또한 ACL(Agent Communication Message)와 KQML(Knowledge Query and Manipulation Language)간의 파싱 기능과 Push, Pull, Immediate 등 다양한 메시지 타입을 제공한다.

4. 구현 및 평가

제안된 미들웨어의 검증을 위해 AIM 프로토타입을 구현하였다. 또한 이를 사용해 사용자 중심 서비스 제공 과정을 실험하였다. 다음 (그림 7)은 구현 환경을 보인다.

구현에 사용된 모든 컴퓨팅 기기는 각각의 하드웨어에 맞는 OS와 자바 가상 머신(Java Virtual Machine, JVM), 에이전트 플랫폼, ICS 등을 사용한다. 각각의 OS와 자바 가상 머신은 컴퓨팅 기기 사용을 위한 기초 플랫폼이 되며, 에이전트 플랫폼은 지능형 에이전트의 동작을 지원한다. 그리고 ICS는 각 플랫폼과 에이전트, HCI 컴포넌트 간의 유연한 통신을 지원한다.

AIM 도메인 서버는 HCI 컴포넌트를 통해 도메인 내 사

용자 중심 서비스의 주체가 되며, 내부의 HCI 컴포넌트를 통해 사용자 중심 서비스를 위한 상황 정보 인식, 사용자 행동 추론, Service Agent 생성, UAML 생성 등을 담당한다.

위치 센서는 RFID 센서의 로우 데이터(Raw Data)를 받아 AIM에서 사용할 센싱 데이터로 변환한다. 위치 센서의 Device Agent는 위치 센서에 고정되어 RFID 센서와의 통신을 통해 로우 데이터를 전송 받고, 이를 바탕으로 사용자의 위치를 나타내는 센싱 데이터로 변환한다. 변환된 센싱 데이터는 AIM 도메인 서버에게 보내져 사용자 위치 추론에 사용된다.

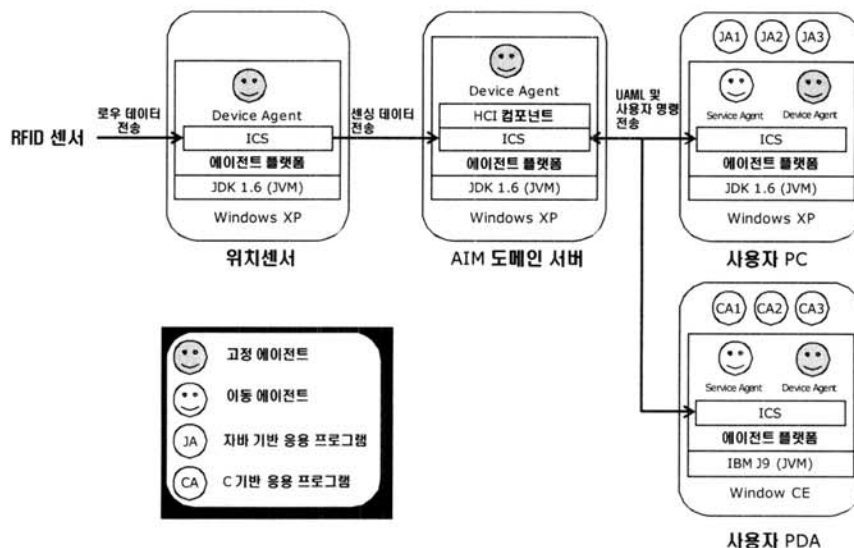
사용자 PC와 PDA는 사용자와 상호작용한다. 이를 위해 각 플랫폼에는 자바와 C로 구현된 응용 프로그램들(JA,CA)이 설치된다. PDA의 경우, 부족한 컴퓨팅 자원을 고려해 PDA에 적합한 Embedded Visual C++ 4.0을 사용하여 응용 프로그램(CA)을 구현하였다. 일반 PC의 경우, 이러한 제약사항이 없으므로 자바를 사용하여 응용 프로그램(JA)을 구현하였다. 각각의 플랫폼에 있는 Device Agent는 응용 프로그램(JA,CA)들을 제어한다. AIM 도메인 서버에서 생성한 Service Agent는 Device Agent를 통해 응용 프로그램들을 제어함으로써 사용자에게 UI를 제공한다.

4.1 사용자 중심 서비스 시나리오

본 절에서는 AIM을 이용하는 사용자 모달리티 기반 서비스 시나리오들을 보인다. 각 시나리오는 현재 가장 활발하게 연구되고 있는 유비쿼터스 스마트 공간을 대상으로 작성되었다.

4.1.1 시나리오 I : 사용자 위치 기반 적응형 UI 제공 기술

사용자가 연구실 중앙의 프로젝트로 이동한다. 위치 센서가 사용자의 위치를 인식한다. 위치 센서는 인식한 사용자의 위치 정보를 AIM 홈 서버에 전달한다. AIM 홈 서버의 CAS는 개미 군집화 알고리즘을 통한 센싱 정보의 변화 여부를 추론하여, 사용자의 위치 변화가 유효하다 판단할 경



(그림 7) 구현 환경

우, 사용자의 위치 이동에 대한 IC를 생성한다. 이를 통해 AIM 홈 서버는 사용자의 프로젝터 접근을 인식한다. AIM의 UCAS는 이러한 사용자의 위치 변화를 인식하여 사용자의 프로젝터 사용 요구를 추론한다. 그리고 UCAS는 사용자에게 프로젝터 사용을 위한 적응형 UI를 제공하는 서비스를 수행하는 Service Agent를 생성한다. Service Agent는 UAML로 기술된 프로젝터 UI를 PDA로 가져와 PDA의 UIS를 통해 표시한다.

4.1.2 시나리오 II : 사용자 이동에 따른 이동 에이전트를 이용한 UI 이주 기술

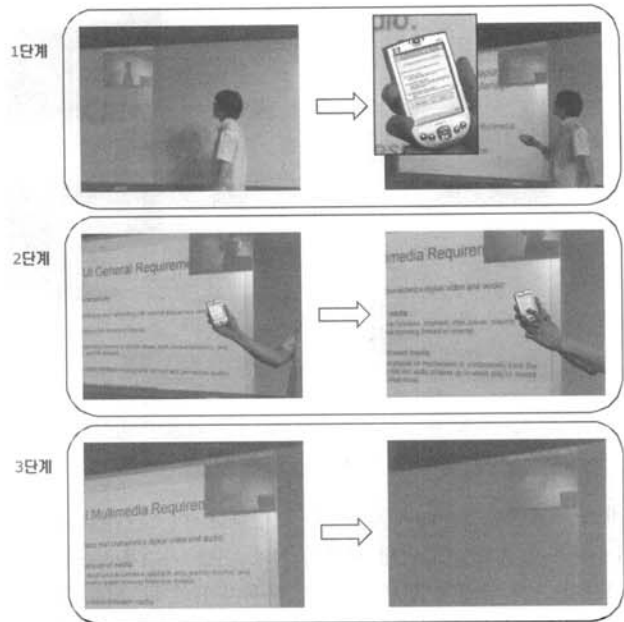
연구실 내의 사용자 A, B는 각자의 컴퓨터를 사용하여 작업한다. 이때 작업하는 프로그램은 본 논문의 구현을 위해 제작한 프로그램으로 제한한다. 각자의 작업물에 대한 논의를 위해, 사용자 B는 작업 중인 자신의 프로그램에서 이동 버튼을 클릭한다. 사용자 B의 컴퓨터에 존재하는 AIM은 사용자 B의 이동 버튼 클릭 명령을 감지하고, UI의 컨텍스트와 데이터를 UAML로 기술하여 UI 이주를 준비한다. 사용자 B는 A의 컴퓨터로 이동한다. 시나리오 I과 같이 홈 서버의 AIM은 위치 센서를 통해 사용자 B가 사용자 A의 컴퓨터로 이동함을 인식한다. 사용자 B의 이동에 대한 인식을 통해 사용자 A의 컴퓨터에 존재하는 AIM은 이동한 사용자 B의 UI를 가져올지 묻는다. 확인을 누르면, 사용자 B의 UAML 정보가 사용자 A의 컴퓨터로 이동한다. 사용자 A의 컴퓨터는 UAML로 기술된 UI의 컨텍스트와 코드를 받아 활성화하고 이를 표시한다.

4.2 구현 결과

본 절에서는 제안된 미들웨어의 검증을 위해 구현한 AIM 프로토타입의 사용자 중심 서비스 제공 시나리오 시현을 보인다. 시나리오 시현에 대한 과정과 시현에 있어서의 AIM 프로토타입의 동작을 설명한다. 다음 (그림 8)은 첫 번째 시나리오, 사용자 위치 기반 적응형 UI 제공 기술의 시현 과정을 보인다.

(그림 8)의 1단계에서 AIM 도메인 서버의 프로젝터 UI 제공을 보인다. AIM 도메인 서버는 RFID 위치 센서를 통해 사용자의 프로젝터로의 이동을 인식한다. 이러한 사용자의 위치 변화는 사용자에 대한 상황 정보, IC이다. AIM 홈 서버는 컨텍스트 정보 IC를 바탕으로 사용자의 프로젝터 사용 요구를 추론한다. 사용자의 프로젝트 사용 요구를 충족시키기 위해 AIM 홈 서버는 Service Agent를 생성한다. 이 Service Agent는 이주 및 UIS 이동을 통해 PDA에 프로젝터 UI를 활성화 시킨다.

(그림 8)의 2단계는 사용자의 UI 이동을 보인다. 사용자는 PDA를 통해 1단계에서 생성된 프로젝터 UI를 조작하여 프로젝터의 표시 내용을 변경할 수 있다. 사용자의 프로젝터 UI 조작에 대한 메시지는 AIM 도메인 서버에게 전송된다. AIM 내 CAS의 MPC는 이 메시지를 수신하여 IC를 생성한다. AIM 도메인 서버는 사용자의 프로젝터 조작 요구를 충족시키기 위해 Service Agent를 생성한다. 이 Service



(그림 8) 시나리오 I : 사용자 위치 기반 적응형 UI 제공 기술

Agnest는 사용자 주변에 위치한 프로젝터로 이동하여 프로젝터의 표시 내용을 변경한다.

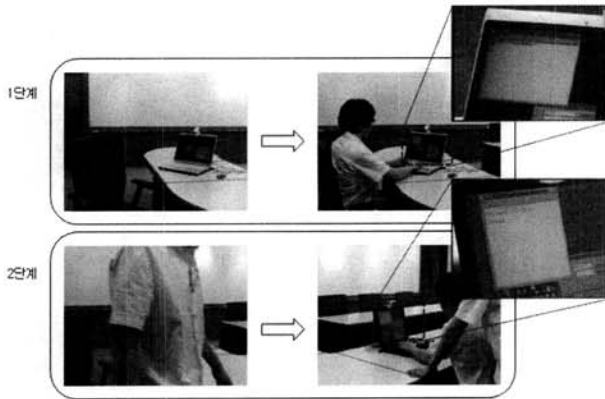
(그림 8)의 3단계는 AIM 도메인 서버의 프로젝터 자동 비활성화를 보인다. AIM 도메인 서버는 RFID 위치 센서를 통해 사용자가 다른 위치로 이동함을 인식한다. 이에 따라 1단계에서 인식한 사용자의 프로젝터 사용 요구를 소거하고 프로젝터는 자동적으로 비활성화 되어 아무 내용도 표현하지 않게 된다.

(그림 8)은 사용자 위치 기반 적응형 UI 제공 기술의 시현을 세 단계로 보였다. AIM 도메인 서버는 센서를 통한 상황 정보 인식, 사용자의 요구 추론, 서비스 수행 과정을 통해 동작한다. 세 단계 모두 이러한 AIM 도메인 서버의 수행 과정을 통해 동작하였다. 이때 CAS는 사용자의 위치 정보와 프로젝터 조작 명령 등 상황 정보를 인식하여 IC를 생성 하였다. UCAS는 생성된 IC를 인식하여 사용자의 요구 추론을 수행하고, 이를 처리할 수 있는 Service Agent를 생성하였다. Service Agent는 DS와 UI를 통해 각각 장비 사용과 UI 제공을 위한 정보 및 연산을 제공 받고, 다른 장치의 Device Agent와 통신하여 서비스를 수행하였다.

다음 (그림 9)는 두 번째 시나리오, 사용자 이동에 따른 이동 에이전트를 이용한 UI 이주 기술의 시현 과정을 보인다.

(그림 9)의 1단계는 사용자 B의 UI 사용을 보인다. 먼저 시나리오 I의 1단계와 AIM 홈 서버는 사용자 B의 컴퓨터 B로의 이동에 따른 컴퓨터 사용 요구를 감지하고, 사용자 B가 과거 사용했던 UI를 재활성화 한다. 사용자 B는 다시 활성화된 UI에 추가적인 조작을 수행한다.

(그림 9)의 2단계는 사용자 B의 이동에 따른 UI 이주를 보인다. 사용자 B는 작업 중인 자신의 UI에서 이동 버튼을 클릭하고 사용자 A의 컴퓨터로 이동한다. 이 이동은 위치 센서를 통해 AIM 도메인 서버에 감지된다. AIM 도메인 서버는 사용자 B의 사용자 A와의 협업 의도를 추론한 후, 서



(그림 9) 시나리오 II : 사용자 이동에 따른 이동 에이전트를 이용한 UI 기술

비스를 제공할 Service Agent를 생성한다. 이 Service Agent는 사용자 A의 컴퓨터에 사용자 B의 UI를 가져올지 묻는다. 사용자 A가 긍정의 답변을 주면, 사용자 B의 UAML 자료가 사용자 A의 컴퓨터로 이동하여, 사용자 A의 컴퓨터에서 UAML의 기술에 따라 UI와 컨텍스트가 표시된다.

(그림 8)과 (그림 9)는 구현된 AIM 프로토타입의 시나리오 I와 II 시현을 보였다. (그림 10)은 이러한 시나리오 시현 과정에서 동작한 AIM 프로토타입의 여러 구성요소들의 UI를 보이고, 이를 바탕으로 그 동작을 설명한다.

(그림 10)의 (A)는 RFID센서로부터 수신한 센싱 정보를 받아 AIM에게 전달하는 위치 센서의 모습을 나타낸다. (B)는 도메인 내에 최상위 서비스를 제공하는 AIM 홈 서버의 UI이다. (B)는 AIM의 동작 과정 중 도메인 내 센서 및 기타 플랫폼을 인식하고 연결을 구축하는 모습이다. (C)는 (B)에서 보여준 센서 및 기타 플랫폼 인식을 수행한 후, 센서와 기타 플랫폼이 전송하는 위치 정보와 사용자의 요구를 수신하여 이들을 추론한 뒤 위치 서비스를 제공하는 과정을 보인다. 마지막으로 (D)를 통해 AIM의 상황정보 인지 기능과 UI 생성 기능, Service Agent를 통한 장비 제어 기능 등의 구현 결과를 확인할 수 있다.

4.3 제안 미들웨어의 기능성 분석 및 평가

다음 <표 1>은 본 논문에서 제시하는 AIM의 기능성과 현재 존재하는 HCI 미들웨어에서 제시하는 기능성들을 각

<표 1> 기존 HCI 미들웨어들과 제안 미들웨어간의 기능성 평가

	Thin-Client	HIC	CAPNET	CAMA	AIM
멀티모달리티 지원	L	H	M	H	H
디바이스 적응적 인터페이스 지원	H	H	H	L	H
사용자 적응적 인터페이스 적용	H	H	M	L	H
상호운용적 통신	L	L	L	M	H
온톨로지 서비스 지원	L	L	L	H	H
Context Sensitive	H	H	M	M	L

항목에 따라 비교 평가한 결과를 보여준다. 평가 기준은 HCI 미들웨어 각 요구사항을 만족시키는 정도에 따라 높음(H, High), 중간(M, Middle), 낮음(L, Low)으로 평가하였다.

- 멀티모달리티 지원여부가 첫 번째 평가 기준이다., 사용자가 가진 모달리티를 사용 및 조합하여 사용자와의 상호작용을 지원하는 것이 HCI 미들웨어의 큰 특징을 평가한다.
- 디바이스 및 사용자 적응적 인터페이스의 지원여부가 다음 평가 기준이다., 다양한 디바이스의 특징과 사용자의 여러 특징 사항을 고려한 사용자 인터페이스의 제공 여부를 평가한다. 또한, 이러한 적응적 사용자 인터페이스의 형성에 필요한 디바이스 및 사용자의 특징 정보를 정형화된 형태로 수집, 유지, 갱신하는 등의 관리 기능을 수행 할 수 있어야 한다.
- 상호운용적 통신의 지원 여부 역시 중요하다. HCI 미들웨어는 이기종의 디바이스 및 미들웨어 시스템 간의 상호변환 통신기술을 지원할 수 있어야 한다.
- HCI 미들웨어는 온톨로지 서비스를 제공하여야 한다. 이를 통하여 앞서 언급된 이기종 디바이스 간의 통신 시 의미론적 상호운용성을 제공함으로써, 실세계에 존재하는 사물들의 개념과 그들 사이의 관계 정의를 통하여 보다 폭넓은 상호작용 서비스를 제공할 수 있게 된다.
- 마지막으로, HCI 미들웨어는 디바이스, 사용자에 직접



(A) AIDE 위치 센서

(B) 도메인 내 센서 및 기타 플랫폼 인식에 대한 AIDE 서버 UI

(C) 사용자의 위치 변화 인식과 추론을 바탕으로 한 위치 서비스 제공

(D) 적응형 UI기술 구현 결과

(그림 10) 사용자 모달리티 기반 적응형 UI 서비스 구현 결과

연관된 정보 이외에도 상호작용에 필요한 여러 가지 상황 정보들을 수집 및 분석하여 더욱 자연스러운 상호작용 서비스를 제공할 수 있어야 합니다. 이를 context sensitive 항목으로 평가하였다.

각 미들웨어에 대한 평가 내용은 다음과 같다.

- 일본 와세다 대학의 썬 클라이언트 기반 미들웨어는 사용자의 현재 상황과 선호도 및 디바이스에 따른 인터페이스 적응을 지원하고 있으나, 사용자가 가지는 다양한 모달리티를 조합할 수 있는, 즉 멀티모달리티를 지원하는 기능을 제시하지 못하였다.
- THALES Research & Technology France의 HIC 미들웨어는 응용, 상호작용, 프리젠테이션 로직 간의 중개 역할을 하는 다양한 에이전트들로 구성되는 아키텍처로서, 다양한 모달리티에 적응하여 서비스를 제공하는 구조를 가지고 있다. 또한, 디바이스 및 사용자 적응 인터페이스 지원이 가능하며, UI context, dialogue history와 같은 상호작용 자원(interaction resource)에 대한 실시간 접근을 보장한다.
- 핀란드 Oulu 대학의 CAPNET 미들웨어는 상황인지 퍼베시브 네트워크 환경에 존재하는 다양한 기기종 디바이스에 적응적인 인터페이스를 제공하기 위한 아키텍처이다. 사용자의 선호도와 현재 컨텍스트 및 디바이스 적응적 인터페이스 기능을 지원하고 있지만, 사용자 컨텍스트와 디바이스 정보를 유지하고 관리하는 정형화된 방법을 제시하지 않았다.
- 스위스 Fribourg 대학의 CAMA 미들웨어 아키텍처는 멀티모달리티와 상황인지를 초점으로 하여 설계된 아키텍처이다. Resource, situation, plan, context history 등의 정보에 대한 온톨로지를 유지하며 다양한 컨텍스트 정보를 관리한다. 또한, data 온톨로지를 통하여 에이전트 사이의 의미론적 통신에 대해서 언급하고 있다. 그러나 이러한 다양한 온톨로지를 이용한 컨텍스트 정보를 통한 디바이스 및 사용자 적응적 인터페이스 지원에 대해 제시하지 않았다.

본 논문에서 제시하는 AIM은 HCI 미들웨어의 요구사항에 대한 분석을 통해 설계된 아키텍처로서, 사용자의 모달리티를 지원하기 위한 기본 컴포넌트 구조와 확장 구조를 제시하고, 사용자 및 디바이스 정보의 관리를 통해 인터페이스 및 시스템 상황에 따른 적응을 지원하도록 설계되었다. 제안된 HCI 미들웨어를 통하여 멀티모달리티를 지원하는 사용자 및 디바이스 적응적인 사용자 인터페이스의 제공이 가능하며, 기기종의 디바이스, 응용, 또는 시스템 사이의 의미론적 상호운용성을 제공할 있다. 하지만 상황 정보 수집에 있어 여러 종류의 센서들로부터 수집되는 다양한 상황 정보에 대한 연구 및 고려가 부족하여, 이에 대한 정보 수집 및 검증이 필요하다.

5. 결 론

본 논문에서는 HCI 미들웨어 요구사항을 분석하고, 이를 기반으로 서비스, 응용 및 사용자 간의 자연스런 상호작용을 지원하는 에이전트 기반 지능형 HCI 미들웨어, AIM을 제안하였다. 제안 미들웨어 아키텍처는 XML 기술을 이용하여, 사용자, 장치, 이벤트와 상황 프로파일을 관리함으로써, 사용자 중심 서비스를 위한 여러 정보들을 관리하였다. 그리고 사용자의 다양한 모달리티 지원과 상황 정보 추론 기능을 제공함으로써 사용자와 컴퓨팅 시스템간의 자연스러운 상호작용을 지원하였다. AIM은 유비쿼터스 환경에 존재하는 다양한 기기종의 디바이스, 미들웨어, 그리고 응용 간의 상호운용성을 지원할 수 있을 뿐만 아니라, 시스템의 하부 구조로서 고정(fixed) 및 이동(mobile) 에이전트 시스템을 지원하여 다양한 목적을 가진 지능형 에이전트의 구현을 지원하였다.

나아가 본 논문에서 제시하는 서비스 시나리오에서는 스마트 공간에서 사용자의 모달리티 기반 디바이스와 서비스에 대한 더욱 손쉽고 자연스런 제어 방법을 제시하였다.

또한, AIM을 이용하여 스마트 공간에서의 사용자 모달리티 기반 장비 및 UI 제어 시나리오를 구현하였다. 이러한 구현을 통해 유비쿼터스 컴퓨팅 환경에서 사용자 개입을 최소화한 사용자 중심 서비스의 가능성을 보임으로써, AIM의 이점을 보였다.

향후 연구과제로서 본 미들웨어들의 세부 기능을 보완하고, 상황인지(context aware), 적응성(adaptive), 반영성(reflective)을 제공하는 타 유비쿼터스 미들웨어와의 상호연동을 고려할 것이다. 또한 다양한 도메인에 AIM을 적용해 봄으로써, 유비쿼터스 환경에서의 HCI 미들웨어의 역할과 연구방향을 제시하는 연구를 수행할 것이다.

참 고 문 헌

- [1] A. Dix, J. Finlay, G. Abowd, and R. Beale, Human-Computer Interaction 2nd Edition, Prentice Hall, 1998.
- [2] ACM SIGCHI, <http://sigchi.org/cdg/cdg2.html>
- [3] http://en.wikipedia.org/wiki/Human-computer_interaction
- [4] T. Nakajima, "A Middleware Component Supporting Flexible User Interaction for Networked Home Appliances," ACM SIGARCH Computer Architecture News, Vol.29, Issue 5, 2001.
- [5] C. Sedogbo, P. Bisson, O. Grisvard, and T. Poibeau, "Human-system Interaction Container Paradigm," Proc. of the Human-Computer Interaction International Conference, 2003.
- [6] P. Repo and J. Rieki, "Middleware Support for Implementing Context-Aware Multimodal User Interfaces," Proc. of the 3rd International Conference on Mobile and Ubiquitous Multimedia, 2004.
- [7] M. Tran, B. Hirsbrunner, and M. Courant, "A Context-

Aware Middleware for Multimodal Dialogue Applications with Context Tracing," Proc. of the 3rd International Workshop on Middleware for Pervasive and ad hoc Computing, ACM International Conference Proceeding Series, Vol.115, 2005.

- [8] FIPA Agent Management Specification, <http://www.fipa.org/specs/fipa00023/SC00023K.html>
- [9] G. S. Kim, J. Kim, H. Cho, W. Lim, and Y. I. Eom, "Development of a Lightweight Middleware Technologies Supporting Mobile Agents," Proc. of the Eighth Pacific-Rim International Workshop on Multi-Agents, 2005.



장 현 수

e-mail : jhs4071@ece.skku.ac.kr
 2002년 성균관대학교 전자전기컴퓨터공학과 (학사)
 2005년 성균관대학교 전자전기컴퓨터공학과 (공학석사)

2006년~현 재 성균관대학교 전자전기 컴퓨터공학과박사과정
 관심분야: 유비쿼터스 컴퓨팅, 이동 에이전트, HCI, 미들웨어 등



김 연 우

e-mail : darobil@ece.skku.ac.kr
 2007년 성균관대학교 컴퓨터공학과(학사)
 2009년~현 재 성균관대학교 전자전기컴퓨터 공학과 석사과정
 관심분야: 운영체제, 알고리즘, 유비쿼터스 등



최 정 환

e-mail : themars@ece.skku.ac.kr
 2007년 성균관대학교 컴퓨터공학과(학사)
 2007년~현 재 성균관대학교 휴대폰학과 석사과정
 관심분야: 추천자 시스템, 상황 인지, 미들웨어, 접근 제어 등



강 동 현

e-mail : kkangsu@ece.skku.ac.kr
 2007년 한국산업기술대학교 컴퓨터공학과 (학사)
 2007년~현 재 성균관대학교 전자전기 컴퓨터공학과 석사과정
 관심분야: HCI, 이동 에이전트, 파일 시스템 등



송 창 환

e-mail : eerien@ece.skku.ac.kr
 2008년 성균관대학교 컴퓨터공학과(학사)
 2009년~현 재 성균관대학교 전자전기컴퓨터공학과 석사과정
 관심분야: 유비쿼터스 컴퓨팅, HCI, 이동 에이전트 등



엄 영 익

e-mail : yieom@ece.skku.ac.kr
 1983년 서울대학교 계산통계학과(학사)
 1985년 서울대학교 전산학과(이학석사)
 1991년 서울대학교 전산학과(이학박사)
 2000년~2001년 Dept. of Info. and Comm. Science at UCI 방문교수

2005년~현 재 한국정보처리학회 학회지 편집위원장
 1993년~현 재 성균관대학교 정보통신공학부 교수
 2007년~현 재 성균관대학교 정보통신처 처장
 관심분야: 분산 컴퓨팅, 이동 컴퓨팅, 이동 에이전트, 시스템 보안, 운영체제, 내장형 시스템 등