

지능형 서비스를 위한 상황인식 해석 구조 설계 및 구현[☆]

Design and Implementation of Context-awareness Interpretation Structure for Intelligent Service

임 재 현*
Jae-hyun Lim

요 약

본 논문에서는 지능형 서비스를 제공하는데 필수적인 상황인식 해석 구조를 제안한다. 제안된 모델은 외부 상황 정보와 내부 상황 정보를 통합한 해석 구조를 가지고 있으며, 상황 정보와 응용 서비스간의 독립성이 유지되도록 설계하였다. 제안된 해석 구조의 검증을 위해서 기존 강의실에 상황인식 개념을 도입한 테스트베드(Ubi-Class)를 구축하고 시나리오를 구성하였으며, 상황 해석에 따라 지능형 서비스가 수행되는지 실험하였다.

Abstract

In this research paper, We have proposed an interpretation structure concerned with context-awareness which is essential to provide intelligent service. The proposed model has an interpretation structure that integrated information of both external and internal context, and been designed to keep independency between context information and application service. To verify the interpretation structure that proposed, we established a test bed (Ubi-Class) in which the concept of context awareness has been introduced to the existing class room, and constituted a scenario, and tested whether intelligent service is performed according to context interpretation.

☞ Keyword : 상황인식(context-awareness), 유비쿼터스(ubiquitous), 해석 구조(interpretation structure), 지능형 서비스(intelligent service)

1. 서 론

1.1 연구 배경

상황인식 컴퓨팅은 인간 세계의 의사소통과 거의 동일한 수준에서 인간과 컴퓨터간의 의사소통이 가능하도록 한다는 동기와 목표에서 출발하고 있다. 즉 인간은 자신의 생각과 의사를 타인에게 전달하는 과정에서 다양한 수단을 적절하게 사용함으로써 매우 효과적인 의사소통을 하고 있다. 인간은 서로 대화를 나누는 도중 함축적인 상황

정보(제스처, 표정 등)를 사용하여 대화의 폭을 넓일 수 있다. 하지만 현재 이러한 인간의 의사 전달 능력은 인간이 컴퓨터와 상호 작용하는 과정에서 제대로 적용되지 못하고 있다. 전형적인 대화형 컴퓨터 기술의 경우 컴퓨터에 제공할 입력 방식은 빈약한 수준이다. 이러한 상황 정보에 대하여 컴퓨터가 보다 용이하게 접근하여 이해하고 또한 이를 적절히 사용하도록 한다면 인간 - 컴퓨터 간의 상호작용에 있어 대화의 수준을 향상시킬 수 있고 결국 이를 기반으로 하여 인간은 보다 유용한 지능형 서비스를 받을 수 있을 것이다[1][2][3].

유비쿼터스 환경 하에서 상황인식에 대한 연구는 국내·외적으로 활발히 연구되고 있지만, 아직 상황인식에 대한 정확한 정의는 확립되지 않았다.

* 종신회원 : 공주대학교 컴퓨터공학부 교수
defacto@kongju.ac.kr

[2006/01/03 투고 - 2006/01/19 심사 - 2006/06/28 심사완료]

☆ 본 연구는 한국과학재단 특장기초연구(R01-2006-000-10555-0) 지원으로 수행되었음

Schilt 등은 ‘상황을 사용자와 오브젝트에 관련된 신원 및 정보’로 정의 하였으며[4][5], Dey 등은 ‘상황을 사용자, 공간, 오브젝트 등의 개체와 관련된 모든 정보’라고 정의하였다. Pascoe는 상황을 ‘특정 관심이 가는 실체의 물리적 개념적 상태의 부분집합’으로 정의 하였다[6]. 이러한 정의들은 응용 서비스가 요구하는 상황을 개별적으로 생성하는 데는 적합하지만, 다양한 응용 서비스에 공동으로 사용하기에는 적합하지 않다[2]. 또한 유비쿼터스 환경 하에서는 여러 가지 센서들과 응용 프로그램들이 사용될 수 있으며, 이들과 통신을 하고 제어를 수행하는 핵심 프로그램이 필요하다. 이 핵심 프로그램은 어떠한 센서, 어떠한 응용 프로그램이 사용되더라도 유동적으로 대응할 수 있어야 하고, 기존 시스템의 변화를 최소화시킬 수 있어야 하며, 적절한 상황인식 해석 구조를 통해 서비스를 실행해야 한다.

싱가포르 국립대에서는 OWL(Ontology Markup Language)기반의 SOCAM (Service-Oriented Context-Aware Middleware)을 발표하였다. 아리노나 주립대에서는 분산 환경을 지원하는 RCSM (Reconfigurable Context-Sensitive Mddlware)을 개발하였다[8][9][10]. 이들 연구는 각각의 응용 서비스에 따라 서로 다른 의미와 형태를 갖는 상황정보를 사용하는 문제가 있다.

미국 조지아 공대에서는 상황인식 애플리케이션에 사용될 수 있는 Context Toolkit 이라는 미들웨어를 개발하였다[6][7]. 이 Context Toolkit은 상황인식 응용 프로그램에 적용이 가능한 모델로써, 이들은 정형화된 상황인식 모델을 가지고 5W1H 라는 개념을 제시하고 있다. 이들 연구는 Who, What, Where, When, Why, How를 사용하여 상황을 해석하지만, 센서와 응용 서비스간의 독립성을 제공하기 위해 상황정보를 관리하는 중간 매개체를 따로 사용해야만 한다.

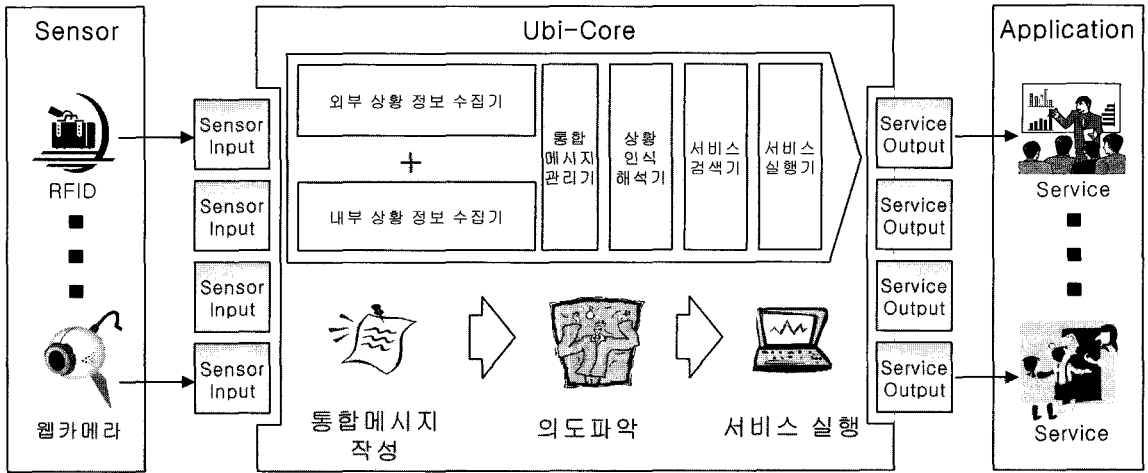
본 논문에서는 유비쿼터스 환경에서 지능형 서비스를 위한 상황인식 해석 구조(Ubi-Core)를 설계하고 구현한다. Ubi-Core는 기존 연구와는 다르

게 센서가 수집한 정보와 사용자의 이력 정보 등 2가지 형태의 상황정보를 통합하여 지능적 서비스를 제공하며, 센서와 응용 서비스간의 독립성을 보장하고 상황 정보의 재사용이 가능하게 설계한다. 이것이 가능한 것은 Ubi-Core는 2가지 형태의 상황 정보를 이용하여 현재 실행시킬 서비스에 필요한 통합 정보를 직접 생성함으로써 응용 서비스에 최적화된 서비스를 제공하기 때문이다. 이러한 통합 메시지를 구성함으로써 서비스 환경의 변화와 센서의 변화 등 다양한 환경에서도 동일한 해석 모델을 가질 수 있으며, 물질공간의 변화에 따른 해석 구조의 변화를 최소화 시킨다.

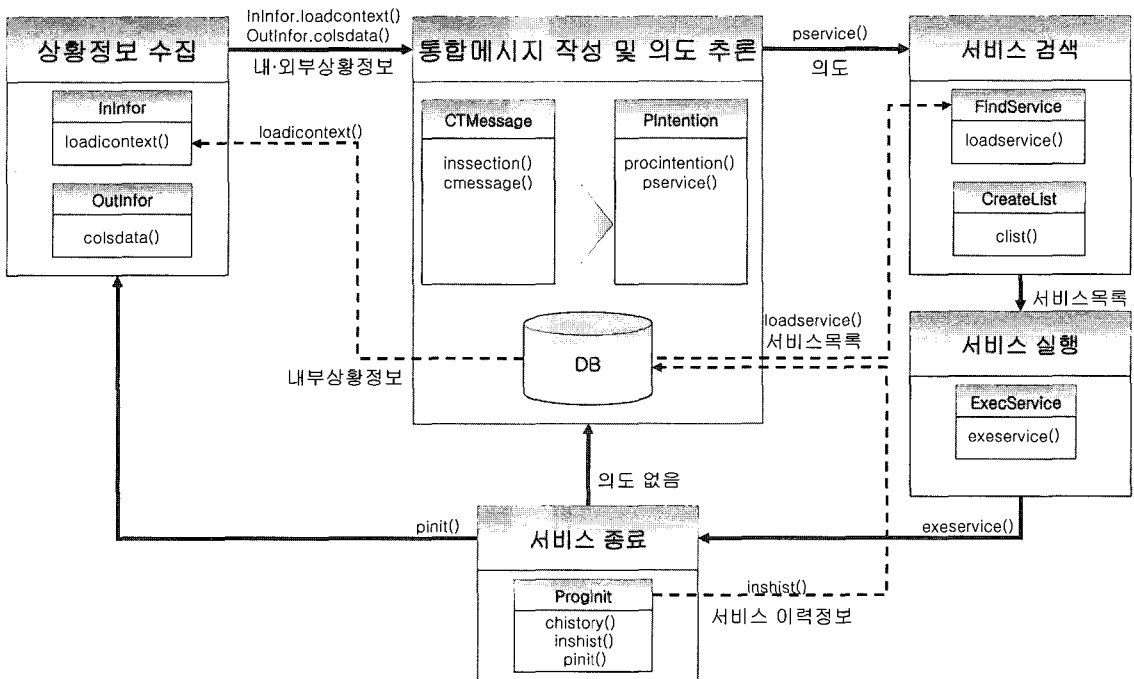
본 논문의 구성은 다음과 같다. 2장에서는 유비쿼터스 환경 하에서의 상황인식 해석 구조를 제안하고, 3장에서는 제안된 모델을 적용한 테스트베드의 구현 상황에 대해 설명한다. 4장에서는 상황인식 해석 구조의 유용성에 대한 실험결과를 분석하고, 5장에서 앞으로의 연구 방향에 대해 살펴본다.

2. 상황인식 해석 구조 : Ubi-Core

관련연구를 기반으로 하여 상황인식 해석 구조를 설계하고 구현한다. Ubi-Core는 센서단과 실제 서비스단의 중간 매개체 역할과 함께 센서관리, 서비스 관리를 통해 최소화된 수정을 통해 어떠한 상황인식 응용에도 사용할 수 있게 설계하였으며, 크게 5개의 구조로 구성하였다. 내·외부 상황 정보 수집기는 직접 센서 및 이력정보 데이터 베이스와 관계를 가지며, 기초 정보를 수집한다. 여러 가지 상황인식 환경에서도 적용이 가능한 상황 정보를 생성하기 위해, 외부 상황 정보는 외부 센서를 통해 상황 정보를 수집한다. 내부 상황 정보는 이력정보와 사용자 입력 정보로써 이전에 이루어진 서비스의 목록, 사용자의 신원 정보 등과 서비스되었던 정보를 기반으로 한 사용자의 취향 등으로 기존의 저장된 정보들이다. 통합 메시지 관리기는 내·외부 상황 정보 수집기를 통해



(그림 1) 상황인식 해석 구조



(그림 2) Ubi-Core의 상태천이도

수집된 기초 정보를 기반으로 통합 메시지를 작성하고, 상황인식 해석기는 통합 메시지를 사용자 의도(Intention)를 파악하는데 사용한다. 서비스 검색기는 사용자 의도에 의한 적절한 서비스들을

검색해 내며, 서비스 실행기는 검색된 서비스를 실행하고 실행에 필요한 컴퓨팅 자원을 관리한다. (그림 1)은 상황인식 해석 구조를 보여주며, (그림 2)는 상황인식 해석 구조의 상태천이도이

다. 상태천이도는 Ubi-Core의 핵심적인 클래스들과 메소드를 포함하며, 각 클래스와 메소드의 역할은 (표 1)과 같다.

(표 1) Ubi-Core의 핵심적인 클래스와 메소드들

역할	클래스명	메소드명	설명
상황 정보 수집	InInfor	loadicontext ()	DB에서 내부 상황 정보를 수집
	OutInfor	colldata()	외부 센서로부터 데이터를 수집
통합 메시지 작성 및 의도 추론	CT Message	inssection()	메시지 섹션에 수집된 상황정보를 바탕으로 데이터 입력
		cmmessage()	메시지 섹션을 바탕으로 통합메시지 작성
	PIntention	prointention()	의도 파악
		pservice()	서비스 검색기에 의도 전달
서비스 검색	FindService	loadservice ()	파악된 의도에 적합한 서비스를 DB에서 검색
	CreateList	clist()	검색된 서비스를 목록화
서비스 실행	Exec Service	exeservice()	목록화 된 서비스를 실행하도록 하드웨어에 명령
서비스 종료	Prognit	chistory()	실행된 서비스를 히스토리로 작성
		inshist()	히스토리를 DB에 이력 정보로 저장
		pinit()	Ubi-Core 초기화

2.1 내부·외부 상황 정보 수집기

외부 상황 정보 수집기는 여러 가지 센서를 통해 객체(Object), 위치정보(Location), 시간정보(Time), 행위정보(Action)의 상황 정보를 수집하고, 내부 상황 정보 수집기는 이력정보, 사용자 입력 등의 상황 정보를 수집하여 서비스에 적합한 통합 메시지를 구성한다. 통합된 메시지는 상황 인식 해석기에 의해 의도를 파악하는데 사용하고,

구축된 통합 메시지는 데이터베이스에 다시 이력 정보로써 저장된다. 상황 정보는 (표 2)와 같다.

(표 2) 상황 정보 라이브러리

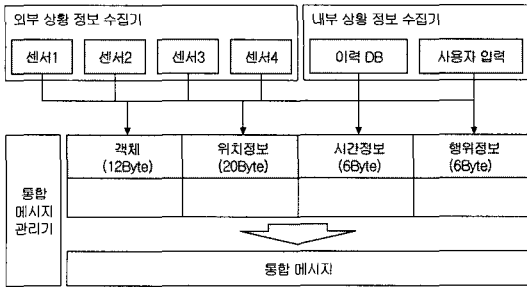
Context	Definition
객체	사용자 혹은 대상에 대한 신원정보
위치정보	객체의 위치정보
시간정보	이벤트 발생 시간 혹은 서비스 개시·개시 예정 시간
행위정보	객체가 취한 의미 있는 행동
의도	객체의 의도

외부 상황 정보 수집기에서 센서들은 서비스 대상 주체와 공간에 의해 변화될 수 있으며, 각 상황 정보에 적합한 센서들은 미리 규정되지 않는다. 예를 들어 객체는 서비스될 사용자 혹은 대상에 대한 신원정보로써, 객체 정보를 수집하기 위하여 RFID나 지문인식기 등을 사용할 수 있다. 상점 등의 매장에서는 RFID를 이용한 고객 신원 관리가 유용하겠지만 중요한 기밀을 다루는 사무실에서는 지문인식기 등이 유용할 수 있다. 혹은 RFID와 지문인식기 등 다수의 센서가 동시에 사용될 수 있을 것이다.

내부 상황 정보 수집기는 이미 입력되어 있는 이력정보와 사용자의 입력에 의한 사용자 정보로 구성된다. 이력정보는 이미 시행된 서비스를 기반으로 구성한다. 이력정보는 데이터베이스에 이벤트가 발생하거나 서비스가 일어날 때 지속적으로 저장되며, 차후 상황인식 해석기를 통해 의도를 유추할 때 사용한다.

2.2 통합 메시지 관리기

수집된 상황 정보들을 (그림 3)과 같이 하나의 통합된 메시지로 구축된다. 외부 상황 정보와 내부 상황 정보를 조합하여 객체, 위치정보, 시간정보, 행위정보의 4가지 상황 정보를 얻어 내며, 이 상황 정보를 토대로 통합 메시지를 구성한다. 통



(그림 3) 통합메시지 작성 구조

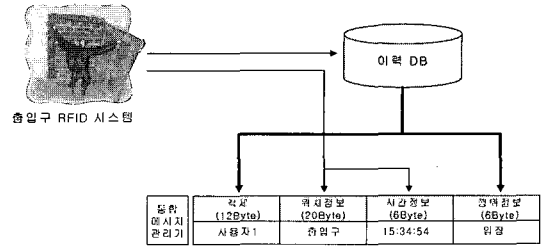
합된 메시지는 상황인식 해석기로 보내지며, 상황인식 해석기는 통합 메시지를 이용하여 사용자의 의도를 유추한다.

메시지 작성 시점은 특정 이벤트의 발생 시점으로 예를 들면, RFID 출입 시스템에서 사용자가 RFID Tag를 소지한 채 입구를 통과했을 때 등으로 주로 외부 상황 정보가 수집되는 시점이 된다. 수집된 외부 상황 정보와 내부 상황 정보를 기초로 작성되는 통합 메시지는 반드시 4개의 구역이 모두 채워져야 할 필요는 없으며, 수집하는 혹은 수집 가능한 정보만이 채워진다. 상황정보에서 시간정보를 제외한 각 라이브러리는 문자형태로 통합메시지 작성에 사용된다. Time은 HH:MM:SS의 시간구조로 작성되어 통합메시지에 추가된다. (그림 4)는 통합 메시지가 작성되는 것을 보여주는 예시이다.

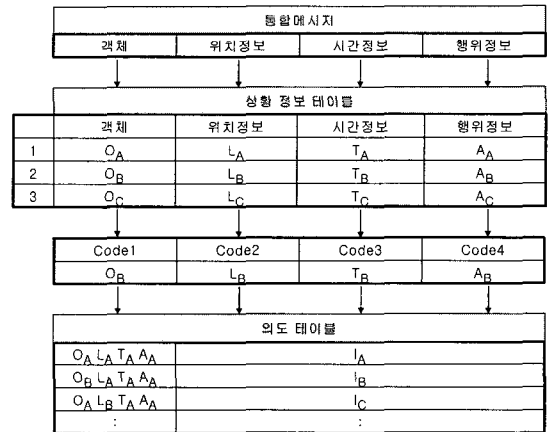
2.3 상황 인식 해석기

통합 메시지 관리기를 통해 작성된 통합 메시지를 토대로 사용자의 의도를 유추해 내는 역할을 한다. 이는 적합한 상황에서의 적절한 서비스를 위한 것으로 통합 메시지의 내용에 따라 해석된다. 상황 인식 해석기는 미리 정의된 코드값을 가지고 있으며, 통합 메시지에 작성된 내용에 맞추어 해당 코드값을 조합한다. 이 코드값에 의해 의도를 유추한다. (그림 5)는 통합메시지에서 의도를 유추하는 과정이다.

데이터 베이스의 상황 정보 테이블은 사용자가 직면할 수 있는 상황에 대한 정보를 담고 있다.



(그림 4) 통합 메시지 작성 예



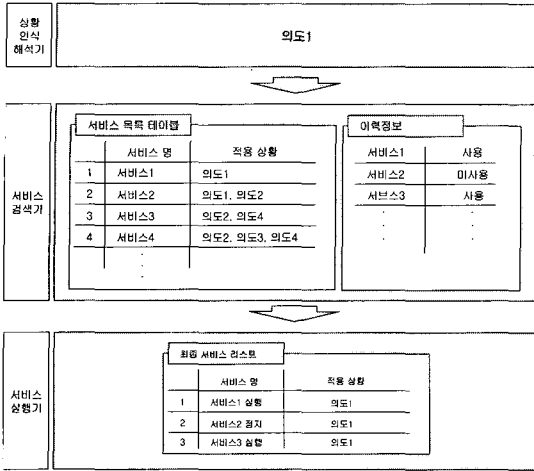
(그림 5) 통합메시지에서 의도 유추 과정

객체는 서비스를 받는 주체들의 신원정보가 담겨 있으며, 위치정보는 상황이 발생할 수 있는 위치의 정보를 담고 있다. 행위정보는 지정된 행위 정보를 담고 있다. 어떠한 서비스가 실행되기 위해 반드시 필요한 사용자의 행위 정보 등이다. 시간 정보는 별도로 데이터 베이스화 되지 않으며, Ubi-core 미들웨어 상의 이벤트 발생 시간 혹은 서비스 예정시간이 계산되어 입력된다.

통합 메시지를 통해 입력된 각 상황 정보는 상황 정보 테이블의 정보와 비교되어 매치되는 정보를 검색한다. 검색된 정보는 지정된 코드값으로 변경되고, 이 코드값을 의도 테이블과 비교하여 해당 상황의 의도를 찾아낸다.

2.4 서비스 검색기와 서비스 실행기

서비스 검색기는 의도에 따라 서비스 될 목록



(그림 6) 서비스 검색 흐름도

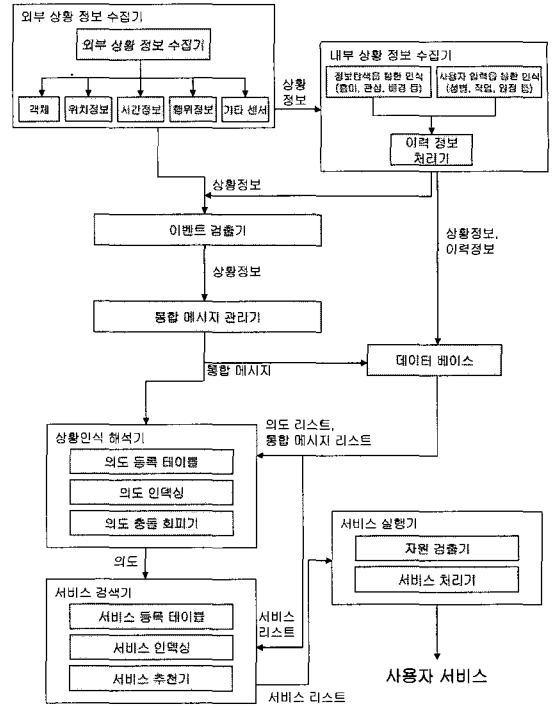
을 검색하고 목록화한다. 검색된 모든 목록은 의도에 따라 실행될 수 있는 모든 서비스를 의미한다. 또한 데이터베이스를 통해 객체의 이력정보를 검색하여 목록화 한다. 검색되어진 이력정보 목록 중에서 객체의 취향에 맞는 서비스를 선택하여 서비스 실행기에 최종 서비스 목록으로 전달한다. 최종 서비스 목록은 객체의 이력정보와 상관없이 항상 실행되어야 하는 서비스를 포함한다.

서비스 실행기는 서비스 검색기에 의해 추천된 서비스 목록에 따라 실제 애플리케이션을 실행한다. 의도와 공간적 상황에 따라 실행되는 서비스는 하나일 수도 있고 다수일 수도 있다. 서비스 검색기는 선택된 서비스들에 따라 해당 서비스의 실행에 필요한 자원이 작동가능 상태에 있는지 자원 검출기에 의뢰하여 실행할 서비스를 최종 선정하여 서비스를 실시한다. (그림 6)은 상황 인식 해석기에서 해석된 사용자 의도에 의해 서비스 목록이 추출되고 실행기로 옮겨지는 과정을 나타낸다.

(그림 7)은 Ubi-Core의 구성요소 사이의 상호관계와 전체 과정을 보여준다.

3. 상황인식 해석 구조 테스트베드

본 논문에서 제안하고 있는 Ubi-Core 모형을



(그림 7) Ubi-core의 구성요소

강의실에 적용하여 구현하였다. 상황인식이 적용된 강의실을 “Ubi-Class”라 하며, 사용된 자원은 RFID, Web 카메라, 압력센서, 조명센서, 빔 프로젝터, 전동스크린, 전등, 강의용 PC등이다. 아래는 Ubi-Class의 시나리오이다.

강사는 RFID Card를 지참하고 강의실에 입장한다. 강사는 강의시작 시간에 맞추어 강의실에 입장한다. 강사가 입장함에 따라 시스템은 강의자원을 위한 자원을 실행하게 된다. 이 때 자원은 강사의 취향에 맞추어 작동하게 되는데 예를 들어 빔 프로젝터를 활용하면서 판서를 하는 강사에게는 빔 프로젝터만을 작동시키게 된다. 하지만 전동스크린도 함께 사용하는 강사에게는 전동스크린을 작동시키는 등 시스템은 강사의 취향에 맞추어 장비를 작동한다. 강의용 PC는 이전에 진행했던 강의에 이어 강의를 진행할 수 있도록 강의 노트를 실행한다. 강사의 입장에 의해 시스템은 강의 진행 모드로 전환된다.

수업을 마친 후 강사는 강의용 PC의 인터페이스를 통해 강의를 종료되었음을 시스템에 통보하고 시스템은 강의 종료 모드로 전환된다. 강의를 종료함에 따라 전등을 모두 켜짐으로 변경하고 빔 프로젝트의 전원을 Off 시키며, 전동 스크린을 올린다.

3.1 Ubi-Class 컴포넌트

Ubi-Class에서는 상황을 인지하기 위해 몇 가지 센서를 사용하였다.

- RFID : Ubi-Class에서는 강사에게 지급되는 카드 형태의 Tag를 사용하였다. RFID Tag는 각각의 중복되지 않는 시리얼 번호를 가지고 있으며, 이 시리얼 번호를 이용하여 신원 확인에 사용하였다. 또한 RFID 안테나와 리더에 의해 제한적인 위치 추적도 가능하다. 사용자가 해당 설치 위치를 지나침으로써 사용자의 위치를 확인하는 것이 가능하며, Ubi-Class에서도 역시 이러한 특징을 이용하여 제한적인 위치 추적의 역할을 RFID 시스템에 부여하였다.
- 웹 카메라 : 웹 카메라는 강사의 위치 추적을 위해 사용하였다. 강사가 강의를 진행하는 위치는 대체적으로 제한적으로 제한된 장소를 촬영함으로써 강사가 강의 위치에 있는지를 확인하는 역할을 수행한다.
- 압력 센서 : 압력을 감지하는 센서로 Ubi-Class에서는 강사의 강사용 테이블 착석유무를 판별한다. 착석유무는 강의용 PC의 제어에 관련된 정보으로써 강사의 착석에 따라 강의용 PC에는 Ubi-Class 인터페이스가 실행된다. 이 인터페이스는 상황인식 기반에서 이루어지는 서비스를 강사가 스스로 선택하여 실행할 수 있도록 해준다.
- 조명 센서 : 전등을 제어하기 위한 센서이다. 각 강사에 따라 강의 환경의 취향이 다

르기 때문에 강의실에 설치된 조명의 사용도 취향에 따라 달라지게 된다. 이러한 조명의 작동 상태를 확인하기 위해 조명 센서를 사용하였다.

각 센서들은 개별적으로 작동하며 상황정보를 Ubi-core에 전달한다. 센서들의 역할은 (표 3)과 같다.

현재 Ubi-Class는 강사에 대한 서비스 지원에 초점을 맞추고 있다. 그에 따라 RFID를 통한 위치 추적은 강사에게만 제한적으로 사용하였다. 신원 정보는 이미 기 구축된 학내 정보망을 이용하며, RFID Tag 시리얼 번호를 통해 신원을 확인하였다.

강사가 강의실에 입장함으로써 강의를 시작한다는 의도는 확인이 될 수 있지만, 그에 따라 원하는 환경을 만들기 위해서는 별도의 서비스가 이루어져야 한다. 이러한 환경적 요소는 여러 가지가 있을 수 있겠지만 Ubi-Class에서는 3 가지 요소로 제한하였고, 그 요소는 빔 프로젝트의 사용유무, 전동 스크린의 사용유무, 전등의 작동 상태이다. 이러한 취향은 이력정보를 통해 유추할 수 있으며, 보다 정확한 판단을 위해서는 미리 강사가 서비스 목록을 선택할 수도 있다.

강의의 시작과 종료는 매우 중요한 사안이며, 높은 정확도를 요구한다. 이는 기본적으로 상황 정보 라이브러리 중 시간정보라는 상황 정보를 이용한다. 강의의 시작시간과 종료 시간은 미리 데이터베이스에 저장되어 있으며, 강사가 강의실에 입장함으로써 강의가 시작된다. 종료는 몇 가

(표 3) Ubi-Class 센서

센서 이름	역할	전달 값
RFID	신원확인, 위치 추적	RFID Tag Serial No.
웹 카메라	위치 추적	촬영된 정지 영상 이미지
압력 센서	위치 추적, Action 추적	On/Off
조명 센서	전등 작동 상태 확인	On/Off

지 경우가 있을 수 있으며, 종료 상황에 조건이 맞을 경우 시스템은 강의를 자동 종료 한다.

3.2 Ubi-Class에서의 상황정보 처리

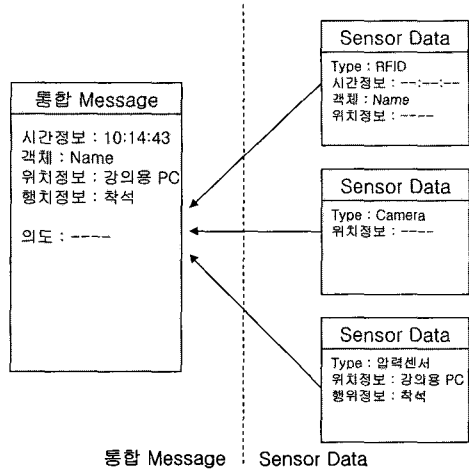
외부 상황 정보 수집기에 의해 각 센서는 수시로 모니터링을 하며, 발생한 상황 정보는 통합메시지 관리기로 전달된다. 통합 메시지 관리기는 수집된 정보를 종합해 통합 메시지를 구축한다. Ubi-Class에서 사용된 센서들이 수집하는 정보는 (표 4)와 같다.

각 블록은 상황 정보 목록과 같이 객체, 위치 정보, 시간정보, 행위정보, 의도로 나누어지며, 통합메시지 관리기에서 초기 통합메시지가 구축되는 시점에서 의도 항목과 수집되지 않은 정보는 제외한다. 예를 들어 강사가 강의용 테이블에 착석하게 되는 경우 RFID를 통해 수집된 객체와 압력센서에 의해 수집된 위치정보와 행위정보, 그리고 시스템으로부터 수집되는 시간정보가 상황 정보로 수집된다. 위치정보에 대한 정보를 수집하는 센서는 총 3개가 있지만 이 시점에서 강사는 강단을 벗어났기 때문에 Web Camera에 의한 정보 수집은 이루어지지 않으며, RFID에 의한 위치 정보 수집도 일어나지 않는다. 대신 압력센서에 의한 위치정보가 사용된다(그림 8). 통합메시지 관리기는 이 4가지 항목으로 초기 통합메시지를 작성하여 상황인식 해석기로 통합메시지를 전달 한다.

외부 상황 정보에서 행위정보는 보다 정확한 사용자의 의도를 유추하기 위해 중요한 요소로

(표 4) 센서에 의해 수집되는 상황 정보 목록

센서 이름	수집 정보
RFID	객체, 위치정보, 시간정보
Web Camera	위치정보, 시간정보
압력 센서	위치정보, 시간정보, 행위정보
조명 센서	행위정보



(그림 8) 통합 메시지 구축의 예(강사 착석)

사용된다. 사용자의 현재 행동에 대한 정보를 행위정보라 정의하는데, 서비스 상에서 사용자가 취할 수 있는 여러 가지 행동을 데이터베이스에 미리 정의해 둔다. 본 시나리오에서 정의된 행위 정보는 (표 5)와 같다.

(표 5) Action 정의

행위정보	수집 센서	행위의 유추 가능한 해석
강사 입장	RFID	강의 시작
강사 퇴장	RFID	강의 종료
강사 착석	무게 센서	메뉴 실행
강사 기립	무게 센서	강의 진행

상황인식 해석기는 통합메시지를 전달받아 현재의 공간적 상황을 파악하고, 이력정보 데이터베이스에서 이력정보를 수집한다. 통합메시지에 수집된 이력정보를 토대로 의도를 유추하고 유추된 의도는 다시 통합메시지의 의도 블록에 저장된다. (표 6)은 통합메시지의 코드값이다. 이 특정한 코드 값은 2진수로 구성되어 해당 통합메시지 구역(객체, 위치정보, 시간정보, 행위정보)의 허용 값이 들어있는지를 판별하여 코드값이 부여된다. 예를 들어 강사가 강의를 위해 강단에 선 경우(그림 9)와 같이 통합메시지 상에 객체(강사), 위치정

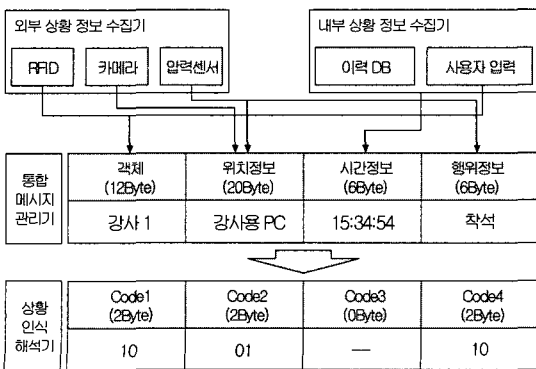
(표 6) 의도 코드 값에 의한 테이블

코드값	상황	정의된 의도	실행 서비스
100101	강사가 강의실에 들어옴	강의 시작	- 강의노트 실행 - 프로젝터 On - 전등 조정 - 전동스크린 Down
100110	강사가 강사용 의자에 앉음	메뉴 실행	- 인터페이스 View
101001	강사가 강단에 위치	강의 진행	- 인터페이스 Hide
010101	강사가 강의실에서 퇴실 또는 입실 전	강의 종료	- 강의노트 종료 - 프로젝터 Off - 전등 All On - 전동스크린 Up

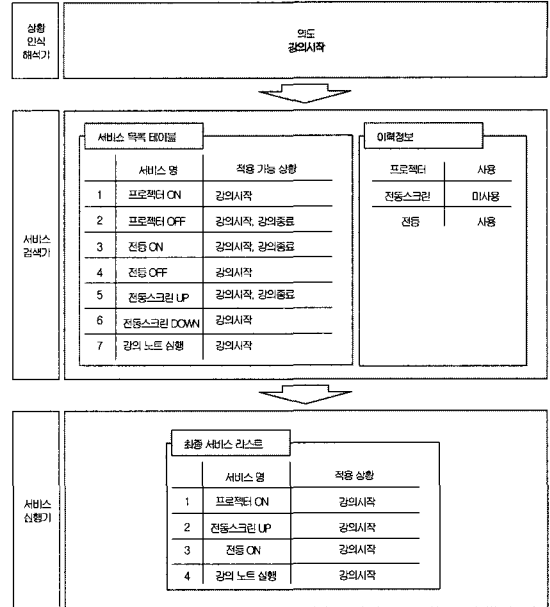
보(강단), 시간정보(현재 시간), 행위정보(착석)으로 코드 조합은 “100110”이 된다. 이중 시간정보의 경우 항상 측정되는 값이기 때문에 코드조합에는 포함하지 않는다.

상황인식 해석기는 재 작성된 통합메시지를 서비스 검색기에 전달하고, 서비스 검색기는 통합메시지를 토대로 서비스 데이터베이스에서 실행 가능한 서비스 목록을 검색한다. Ubi-Class에서 실행 가능한 서비스로는 빔 프로젝터의 작동(On/Off), 전동스크린 작동(Down/Up), 전등 상태변경(All On/All Off/On 1,2,3), 전자출석부 실행, 강의노트 실행, Ubi-Class 인터페이스 실행 등이 있다.

(그림 10)은 강의시작 의도에 따라 서비스 검색



(그림 9) 의도의 유추 (메뉴실행)



(그림 10) 강의시작 의도에 의한 서비스

기와 서비스 실행기가 동작하는 과정을 보여준다. 강의시작 의도에 적합한 서비스들이 데이터베이스에서 검색되어 목록화 된다. 이와 함께 강사의 이력정보가 호출되며, 강사의 이력정보에 따라 서비스 목록 중 실제 실행되어야 할 서비스들이 추천된다. 강사는 강의 시 프로젝터를 사용하고, 전등을 모두 켜다. 반면 전동스크린을 사용하지 않는다. 이러한 이력정보에 따라 서비스 검색기는 프로젝터 ON, 전동스크린 UP, 전등 ON의 서비스를 선택하여 서비스 실행기에 전달한다. 서비스 목록을 전달 받은 서비스 실행기는 자원검출기에게 서비스 실행할 자원의 현재 상태 확인을 의뢰하며, 자원검출기의 응답에 따라 서비스를 실행한다.

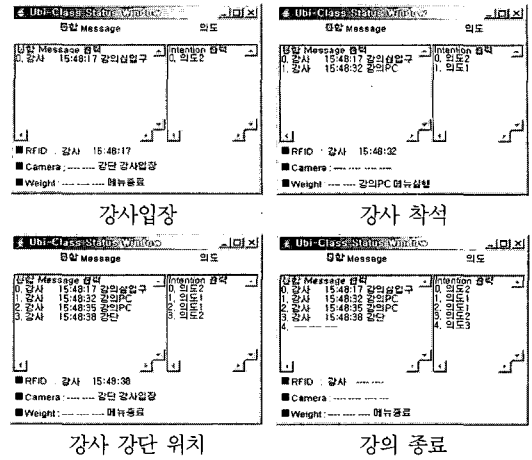
4. 실험

본 논문에서 제안한 Ubi-Core의 실용성과 위에서 열거한 문제점 및 개선점의 효과를 확인하기 위해 (그림 11)과 같이 강의실을 테스트 베드로 하여 실험을 실시하였다.

강의실은 30대의 학생용 PC와 1대의 강의용 PC가 있고, 고정적인 빔 프로젝터와 전동스크린이 설치되어 있다. 또한 전등은 3열 3행으로 9개가 있으며, 각 열별로 조정이 가능하다. 실험에는 이미 빔 프로젝터와 전동스크린을 익숙히 사용하는 강사 5명이 10회씩 실험하였다. 강사에게는 Ubi-Class의 대략적인 개요와 시스템의 실행에 대한 교육을 하였으며, RFID Tag에 대한 간단한 설명과 사용법을 설명해준 상태에서 실험에 임하도록 했다. 또한 Ubi-Class에 사용된 Ubi-core는 의도적으로 각 센서의 상태와 상황의 전달 및 처리를 확인하기 위해 별도로 제작된 Ubi-Class Status Window(UCSW)를 통해 상시 상태를 확인할 수 있도록 하였다. UCSW는 Ubi-core에서 센서와의 통신에 의해 발생하는 이벤트 현황을 출력하고, 그에 따라 작성되는 통합메시지를 출력한다. 또한 파악된 의도를 개발자가 확인할 수 있도록 출력한다. UCSW는 실제 Ubi-core에서는 필요하지 않은 모듈로써 단지 시스템을 구현함에 있어 정확한 결과가 생성되는 확인하기 위한 목적으로 개발하였다.

(그림 12)는 강사가 강단에 위치한 상태로써 강의 진행 상태에서의 통합 메시지와 유추된 의도를 나타낸다. 시나리오에 따라 각 시간별 센서의 동작 상태와 내·외부 상황 정보 수집기에 의해 수집된 데이터를 기반으로 상황을 해석하는 모습을 볼 수 있다.

실험은 실제 강의시간의 시간으로는 실험의 시간소요가 너무 많기에 강의를 진행한다는 시나리오에 따라 임의로 시간을 축소하여 진행하였으며,

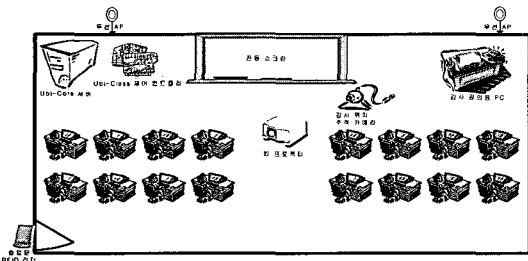


(그림 12) Ubi-Class Status Window

기본 시나리오는 학생이 출석한 후 강사의 출석, 강의시작, 임의의 의도 실행, 강의종료의 순으로 진행하였다. 그 결과 강의자의 의도를 정확히 파악하여 자동화된 서비스를 제공하였다.

실험의 평가를 위해 기존의 강의실에서의 기기 조작의 횟수와 조작에 소요되는 시간, 강의 보조 기기의 실행시간, 출석 확인 소요시간을 확인하여 정량적 평가로 사용하였다. (표 7)은 정량적 평가의 결과를 나타낸다.

기존의 강의실에서는 빔 프로젝터의 실행, 전등의 설정, 전동 스크린 작동, PC 로그인, 강의노트 실행등 강사가 강의를 시작하기 전 입력수가 총 6회로 나타났다. 이에 반해 Ubi-Class에서는 단 1회의 입력(PC 로그인을 위한 입력)만으로 강의를 시작할 수 있었다. 수업의 준비가 완전히 완료되는 시간 또한 Ubi-Class가 기존의 강의실에 비



(그림 11) 강의실 테스트 베드

(표 7) 정량적 평가 결과

평가 항목	Ubi-class 시스템 이용시	기존 강의실 이용시
전체동작 입력 횟수	1번	6번
전체동작 입력시간	1초	9초
전체시스템 평균동작시간	2분 56초	3분 5초
수업준비 평균완료시간 (출석확인포함)	2분 57초	6분 4초

해 50%이상 감소하였음을 알 수 있다. 상기 실험의 시간은 모든 강사에 해당되는 시간이 아닌 평균적인 시간이며, 강사의 취향과 나이, 건강 상태 등 여러 가지 변수로 인해 시간과 입력의 수량은 변화될 수 있다. Ubi-Class에서는 시스템의 오작동이나 갑작스러운 강사의 의도 변화에 따라 오류가 발생할 수 있다. 실제 실험에서는 센서와 직접 연결되고 센서와 서버를 연결해주는 Ubi-Class 제어 컨트롤러에 의한 오류가 발생하였다. Ubi-Class 제어 컨트롤러의 오류는 외부에서 제작된 하드웨어 컨트롤러가 많은 센서를 제어하고 통신하면서 불안정한 전압을 발생함에 따라 다운되거나 정보의 전달이 지연되는 문제점으로 하드웨어 상의 문제로 상황 인식 시스템 자체의 문제는 아니었다. 또한 RFID 인식기의 인식 오류가 발생하기도 했다. RFID의 인식상의 문제는 하드웨어적인 문제로써 테스트 베드구현에 사용된 13.56MHz RFID 시스템은 인식거리가 짧고 Tag와 안테나가 마주하는 방향에 따라 인식율에 상당한 차이를 보였다. 이는 하드웨어에 국한된 문제로 상황 해석은 정상적으로 이루어졌다.

5. 결론 및 향후연구

유비쿼터스 컴퓨팅 환경 하에서 모든 컴퓨터 자원은 유무선의 네트워크로 연결되어 정보의 공유를 이루고 있다. 이들 자원들이 사용자의 지능적 요구를 파악하고 사용자가 처한 상황에 맞는 서비스를 제공하기 위해서는 상황인식 기술이 필요하다. 본 논문에서는 유비쿼터스 환경에서 지능형 서비스를 위한 상황인식 해석 구조(Ubi-Core)를 제안하였다. Ubi-Core는 기존 연구와는 다르게 센서가 수집한 정보와 사용자의 이력 정보 등 2가지 형태의 상황정보를 통합하여 지능적 서비스를 제공하며, 센서와 응용 서비스간의 독립성을 보장하고 상황 정보의 재사용이 가능하게 설계하였다. 이것이 가능한 것은 Ubi-Core는 2가지 형태의 상황 정보를 이용하여 현재 실행시킬 서비스

에 필요한 통합 정보를 직접 생성함으로써 응용 서비스에 최적화된 서비스를 제공하기 때문이다.

제안된 해석 구조의 검증을 위해서 기존 강의실에 상황인식 개념을 도입한 테스트베드(Ubi-Class)를 구축하고 시나리오를 구성하였으며, 상황 해석에 따라 지능형 서비스가 수행되는지 실험하였다. 그 결과 강의자의 의도를 정확히 파악하여 자동화된 서비스를 제공하였으며, 수업을 준비하기 위해 소모되었던 불필요한 시간도 기존 강의실에 비해 50%이상 감소하였다.

본 논문에서 제안된 모델은 상황인식을 위한 기초적인 단계로써 앞으로 몇 가지 연구해야 할 사항이 있다. 우선 센싱 되어진 대량의 정보를 저장하고 보다 빠르게 처리하는 방안이 필요하며, 다중 사용자 환경에서 서비스 대상을 식별하는 기술 연구가 이어져야 한다.

참고 문헌

- [1] Dey, A.K., D. Salber and G.D. Abowd. "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications", Anchor article of special triple issue on Context-Aware Computing. The Human-Computer Interaction (HCI) Journal. Vol. 16, 2001.
- [2] Dey A.K., "Providing Architectural Support for Building Context-Aware Applications", Unpublished PhD Thesis, Georgia Institute of Technology, 2000.
- [3] Bill N. Schilit, Anthony LaMarca, Gaetano Borriello, William Griswold, David McDonald, Edward Lazowska, Anand Balachandran, Jason Hong, and Vaughn Iverson, "Ubiquitous Location-Aware Computing and the "Place Lab" Initiative", The First ACM International Workshop on Wireless Mobile Applications and Services on WLAN (WMASH), Sept. 19, 2003.

- [4] J. Pascoe, "Adding generic contextual capabilities to wearable computers", in Proceedings of 2nd International Symposium on Wearable Computers, pp.92-99, 1998.
- [5] Y. Nishibe et al., "Mobile digital assistants for community support", AAAI Magazine 19(2), pp.31-49, 1998.
- [6] Anind K. Dey, Daniel Salber, and Gregory D. Abowd. A context-based infrastructure for smart environments. In Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99), Dublin, Ireland, pages 114-128, December 1999.
- [7] Dey, A. and Abowd, G., "Towards a Better Understanding of Context and Context-Awareness", Workshop on the what, who, where, when and how of context-awareness at CHI 2000, 2000.
- [8] Ranganathan A, Campbell RH. "A middleware for context-aware agents in ubiquitous computing environments" In: ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil, 2003.
- [9] B. Brumit, B. Meyers, J. Krumm, A. Kern, S. Shafer, "Easyliving: technologies for intelligent environments", Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing, pp.12-.27, 2000.
- [10] S.S. Yau, F. Karim, Y. Wang, B. Wang, S. Gupta, "Reconfigurable context-sensitive middleware for pervasive computing", IEEE Pervasive Comput. 1 (3), pp.33-40, 2002.

● 저 자 소 개 ●



임 재 현

1986년 중앙대학교 전자계산학과 졸업(학사)
 1988년 중앙대학교 대학원 전자계산학과 졸업(석사)
 1998년 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
 1998~현재 공주대학교 컴퓨터공학과 교수
 관심분야 : 상황인식, RFID/USN, 인터넷 기술
 E-mail : defacto@kongju.ac.kr