

## 유비쿼터스 학습공간과 OWL 상황 모델 설계를 통한 상황 인식 학습 시스템 구현

홍명우\*, 이용환\*\*

### Implementation of Context aware Learning System by Designing Ubiquitous Learning Space and OWL Context Model

Myoung-Woo Hong \*, Young-Whan Lee \*\*

#### 요약

유비쿼터스 컴퓨팅 기술은 유비쿼터스 학습 즉 u-러닝의 출현을 이끌며 미래 학교 교육의 개선 방향으로 제시되고 있다. 유비쿼터스 학습 환경에서는 우선 사용자들이 언제 어디서나 학습을 위해 다양한 내장형 컴퓨터들을 활용할 수 있도록 이러한 기기들을 도처에 편재시키고 네트워크를 통한 상호작용을 지원하는 것이다. 다음 단계는 지능적인 학습 서비스를 지원하는 것으로, 상황 정보를 이해하고 이를 공유하도록 상황 지식과 상황 추론이 가능한 상황 인식 서비스를 도입하는 것이다. 그러나 아직까지 실제 학교를 대상으로 상황 인식 서비스를 지원하기 위한 온톨로지 상황 모델 설계 및 적용에 대한 연구결과는 미비한 상태이다. 이에 본 논문은 상황 인식 기반의 유비쿼터스 학습 서비스를 지원하기 위해, 기존 LAN 환경에서 이루어지는 학교 네트워크를 대상으로 USN을 활용한 유비쿼터스 학습 공간을 제안하고, OWL을 사용해 유비쿼터스 학교 환경에 대한 온톨로지 기반의 상황 모델을 설계한다. 또한 제안한 유비쿼터스 학습 공간과 온톨로지 상황 모델이 유비쿼터스 학습 환경에 적합함을 확인하기 위해 유비쿼터스 환경에서의 상황 인식 학습 서비스들을 구현한다.

▶ Keyword : 유비쿼터스 학습, 상황 인식, 온톨로지

#### Abstract

Ubiquitous computing technology makes an impact on the appearance of u-learning and presents an advanced direction of futuristic school education. In ubiquitous learning environments, various embedded computational devices will be pervasive and interoperable across the network for supporting the learning, so users may utilize

• 제1저자 : 홍명우 • 교신저자 : 이용환

• 투고일 : 2011. 03. 26, 심사일 : 2011. 04. 11, 게재확정일 : 2011. 04. 29

\* 우송정보대학 컴퓨터정보계열(Div. of Computer Information, Woosong College)

\*\* 우송정보대학 방송영상디자인과(Dept. of Broadcasting and Visual Media Design, Woosong College)

these devices anytime anywhere. An important next step for ubiquitous learning is the introduction of context-aware learning service that employing knowledge and reasoning to understand the local context and share this information in support of intelligent learning services. However, the existing studies on design and application of ontology context model to support context-aware service in actual school environments are incomplete state. This paper, therefore, suggests a scheme of constructing ubiquitous learning space for existing school network by introducing USN to support context-aware ubiquitous learning services. This paper, also, designs an ontology based context model for ubiquitous school environments which describes context information through OWL. To determine the suitability of proposed ubiquitous learning space and ontology context model, we implement some of context-aware learning services in the ubiquitous learning environments.

▶ Keyword : Ubiquitous Learning, Context aware, Ontology

## I. 서 론

최근 컴퓨팅 기술은 유비쿼터스라는 용어로 집약되는 새로운 질적 도약의 변곡점에 들어서고 있다[1]. 이러한 유비쿼터스 컴퓨팅 기술의 발전으로 최근 학교 교육을 중심으로 유비쿼터스 학습(Ubiquitous Learning) 즉 u-러닝에 대한 관심이 증가되고 있다. 교육인적자원부에서는 2005년 유비쿼터스 기술을 학교 교육 환경에 적용하는 u-러닝 연구학교 사업을 시작하여 현재까지 미래교육 혁신을 위한 다양한 방향을 모색하고 새로운 교육 환경에 대한 예측과 교육적 활용 가능성을 발굴하고자 노력하고 있다. 이와 관련된 유비쿼터스 학습에 관한 연구들로서, 미래 교육의 모습과 추진 중인 u-러닝 시범학교를 분석하고 해외 사례를 분석한 연구[2], u-러닝을 활성화시키기 위한 로드맵에 대한 연구[3], 유비쿼터스 환경에 적합한 미래 교실 구성 방안에 대한 연구[4], 유비쿼터스 기반의 개인 휴대용 학습 단말기 개발에 대한 연구[5], u-러닝 지원 시스템 연구 및 프로토타입 개발[6], 대학에서의 u-Campus 구축에 관한 연구[7] 등이 대표적인 사례이며, 미래 교육의 발전 방향을 모색하기 위해 다양한 각도에서 접근을 시도하였다. 그러나 이러한 연구들은 아직까지 유비쿼터스 학습 개념에 대한 정의 단계를 크게 벗어나지 못하고 있으며, 시범학교 등의 학교 현장에서의 연구들도 주로 모바일 기기의 교육적 활용 예시를 제시할 뿐 구체적인 유비쿼터스 학교 환경 구축 및 학습 지원을 위한 하부구조 등에 대한 연구는 부족한 실정이다.

우리나라 학교의 전산망은 대부분의 학교 시설에서 유선 네트워크에 접속한 PC를 통해 인터넷을 이용하고 있다[2]. 이러한 상황에서 유비쿼터스 기반의 학교 망을 구축하기 위해 학교 현장의 공간을 재배치하고 이미 사용 중인 시설물을 일

시에 변경하는 것은 상당한 투자와 노력이 요구되며, 실질적으로는 쉽지 않다. 학교의 입장에서는 기존의 환경을 개선하는 방향으로 유비쿼터스 학교 망을 구축하는 것이 현실적이라 할 수 있다[8]. 따라서 현 시점에서 학교에서의 유비쿼터스 네트워크 환경 구축에 가장 근접하고 실현 가능한 방법은 기존의 학교 망과 USN(Ubiquitous Sensor Network) 기술을 결합하는 것이다. 이를 통해, 다양한 내장형 컴퓨터들이 도처에 편재되어 있고, 이러한 기기들이 서로 네트워크를 통해 연결되어 상호작용할 수 있게 된다.

유비쿼터스 학습을 지원하기 위한 다음 단계는 학습자의 정보와 주변의 상황 정보를 고려한 상황 인식 서비스 기반의 유비쿼터스 학습 환경을 제공하는 것이다. 현재 학교에서 학습을 지원하는 컴퓨팅 시스템들은 학생과 교사 중심의 메타포로서, 단순히 교사들은 콘텐츠 공급자이며 학생들은 콘텐츠 소비자로서 동작할 뿐이다[9]. 즉 기존의 학습 시스템들은 학습 환경의 계속된 변화를 고려하지 못하기 때문에 학습자 중심의 서비스 제공이 미흡한 실정이다. 상황 인식 기반의 유비쿼터스 학습 환경을 구축하기 위해서는 유비쿼터스 환경에 있는 학습자 및 다양한 컴퓨팅 개체들에 대한 상황 정보를 저장, 관리, 공유할 수 있는 상황 모델의 구축이 필수적이다.

이에 본 논문은 기존의 학교 망에 USN 기술을 도입한 유비쿼터스 학습 공간의 구성 방안을 제시하고, 이 공간에서 이루어지는 학교 수업을 대상으로 온톨로지 기반의 상황 모델을 제안한다. 또한 제안한 유비쿼터스 학습 공간과 온톨로지 상황 모델이 유비쿼터스 학습 환경에 적합함을 확인하기 위해 상황 인식 학습 서비스를 구현한다.

## II. USN 기반 유비쿼터스 학습공간 설계

USN은 필요한 모든 것에 통신 기능이 있는 스마트 RFID

태그나 센서를 부착하고 이를 통하여 사물의 인식 정보를 기본으로 주변의 환경 정보까지 탐지하여, 이를 실시간으로 네트워크에 연결하여 정보를 관리하는 것을 말하는 것이다 [10]. 본 논문은 유무선 네트워크가 가능한 실습실을 대상으로 그 공간에 그림 1과 같이 RFID 시스템, 물리적 센서 및 임베디드 게이트웨이로 구성된 유비쿼터스 학습 공간을 제안한다. 노트북과 PC 등의 전자제품은 실습실에 일반적으로 배치되는 구성요소이며, 상황 인식 학습 시스템과 에이전트는 상황 인식 학습 서비스를 제공하기 위한 프로그램적인 구성요소로서 LAN을 통해 실습실의 상황 정보를 제어한다.

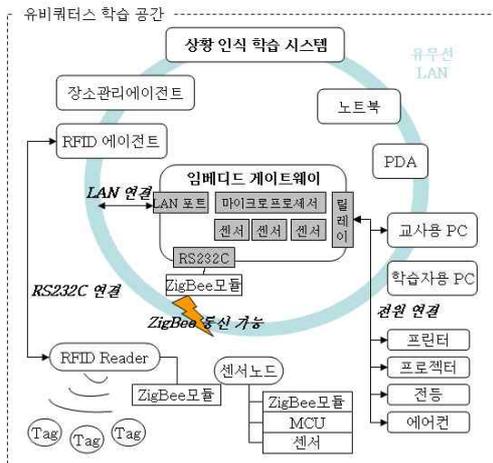


그림 1. 유비쿼터스 학습 공간 구성도  
Fig. 1. Design view of Ubiquitous Learning Space

RFID 시스템은 ISO15693 13.56MHz RFID 표준 프로토콜을 지원하는 RFID 리더 모듈과 13.56MHz RFID 태그를 사용한다. 태그는 교수와 학생들이 소지하며, 리더기와의 태그 인식 거리는 10cm이다. RFID 리더기는 호스트와 RS232C 연결을 통해 시리얼 통신하는데, RS232C에 ZigBee 통신 모듈을 부착하여 무선 통신을 지원할 수도 있다. 실험을 위해 RFID 리더기는 RFID 에이전트가 있는 PC와 RS232C를 통해 연결하고, 파라미터를 115.2 Kbps, 8 Data bit 및 1 Stop bit로 설정하고 ISO15693 명령어를 사용한다.

센서 노드는 CC2420 RF 모듈을 사용하여 2.4GHz ZigBee 통신을 지원하고, 데이터 처리를 위한 MCU로 ATmega128L 프로세서를 사용하며 TinyOS 1.0을 탑재한다. 문의 개폐 센서는 임베디드 게이트웨이의 ZigBee 모듈과 통신하고, 조도센서와 온도/습도 센서는 각각 Hamamatsu사의 S1087과 Sensirion사의 SHT11을 사용하는데, 임베디드 게이트웨이 보드에 직접 부착하여 정보를 얻었다.

그림 2의 임베디드 게이트웨이는 독립된 유비쿼터스 학습 공간에서 동시 다발적으로 발생하는 상황 변화 값의 처리를 전달시키기 위해 자체 제작한 보드로써, 센서 및 전자제품 등의 현재 상태 값을 수집하고 데이터의 프로토콜 변환 및 호스트와의 통신을 수행한다. 이 보드는 마이크로프로세서와 메모리, 주변장치 및 센서들로 구성된다. 마이크로프로세서와 메모리는 데이터의 입출력을 제어하고 데이터의 프로토콜 변환을 담당하는데, UDP 통신과 Ethernet 제어를 위해 RJ45와 ENC28J60을 사용하고, TinyOS, TCP/IP 프로토콜 스택 및 제어 응용을 탑재하기 위해 ATmega128L을 사용하였다[11]. 주변장치로서 RS232C는 ZigBee 모듈을 연결해 두고, 릴레이(Relay)는 PC, 빔 프로젝터, 프린터, TV, 전등 및 문 개폐 장치 등 전자제품들의 전력선과 연결하며, LAN 포트는 유선 LAN에 연결시킨다. 임베디드 게이트웨이는 고유 IP를 가지며 LAN을 통해 센서 에이전트나 장소 관리 에이전트와 통신하게 하였다.

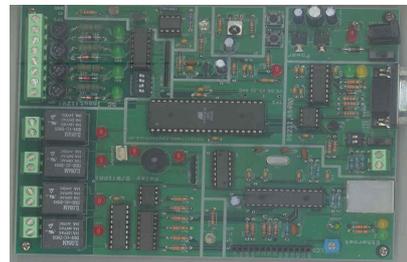


그림 2. 임베디드 게이트웨이 보드  
Fig. 2. Embedded Gateway Board

### III. 학교수업에 대한 온톨로지 상황모델

상황은 개체의 상태를 특징짓는데 사용될 수 있는 정보를 말하는데, 여기에서 개체는 사람, 장소, 또는 물리적이거나 계산적인 객체가 될 수 있다[12]. 이 상황은 사용자나 응용 자체를 포함해서 사용자와 응용 간 상호작용에 관련성이 고려된 것이다. 또한, 상황은 응용 운용 환경의 일부로 응용이 감지할 수 있는 정보를 포함한다[13]. 상황 인식 컴퓨팅은 사용자, 주변 사람과 물체의 집합에 따라 적응적이며, 동시에 시간이 경과하면서 이러한 대상의 변화까지 수용할 수 있는 소프트웨어라고 Schilit에 의해 정의되고 있다[14]. 최근에는 사용자의 작업과 관련 있는 적절한 정보 또는 서비스를 사용자에게 제공하는 과정에서 상황을 사용하는 경우를 상황 인식 서비스로 정의하고 있다. 본 논문에서 다루는 개체는 사용자가 속한 유비쿼터스 학교 공간의 사용자, 위치, 활동, 컴퓨

팅 및 물리적 디바이스 등을 말하며, 이것들에 대한 정보 값과 그 정보들의 변화를 상황 정보라 정의한다. 상황 정보는 상황이 얻어지는 방법에 따라 정적 상황 정보와 동적 상황 정보로 구분할 수 있다. 정적 상황 정보는 사용자, 장소, 물리적 개체 및 컴퓨팅 개체에 대해 그 속성 값을 미리 정의한 정보로서 예를 들면, 어떤 개인의 이름, 전화번호, 이메일 주소 또는 PC의 위치, IP 주소, 식별번호 등을 말한다. 동적 상황 정보는 시간에 따라 변화하는 어떤 값에 대한 정보로서, 물리적 센서로부터 얻어진 온도나 조도의 변화 값이나 RFID 리더기로부터 얻어진 사용자와 그의 위치 정보 등을 말한다.

온톨로지는 공유된 개념화에 대한 정형화되고 명시적인 명세로서[15], 컴퓨터가 처리 가능하도록 구체적으로 표현되어 지식 재활용이 가능한 메타 모델이라 정의할 수 있다. 따라서 온톨로지는 유비쿼터스 컴퓨팅 시스템에서의 상황적 지식 공유와 재사용을 가능하게 하기 위한 일관된 방법을 제공할 수 있다[16]. 본 논문은 W3C의 온톨로지 언어인 OWL-DL(Web Ontology Language-Description Logic)을 이용해 유비쿼터스 학습 환경에 대한 온톨로지 상황 모델을 설계하고 이를 "CULA-ONT(Context aware Ubiquitous Learning Architecture ONTology)"라 한다. 제안한 상황 모델은 owl:Thing 아래에 클래스 46개, 객체 속성 26개, 데이터 속성 24개 및 구성원 33개로 구성되어 있다. 그림 3은 CULA-ONT 온톨로지 상황 모델의 앞부분으로서, W3C의 XML, RDF/RDFS 및 OWL 표준을 따르고 있다는 내용을 정의한 부분이다.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [ . . . ]>
<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2011/2/cula-ont.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  <owl:Ontology rdf:about="">
```

그림 3. CULA-ONT 상황 모델의 일부  
Fig. 3. A Partial CULA-ONT Context Model

1. 클래스의 분류 및 정의

CULA-ONT 상황 모델은 사람, 장소, 활동 및 컴퓨팅 객체를 최상위 개념의 구성요소로 결정하고, 클래스들의 이름은 각각 Person, Place, Activity, ComObject로 명명하였다. 그림 4는 상황 모델의 최상위 클래스와 그 하위 클래스들에 대한 클래스 분류를 그래프를 통해 표현한 것이다. 원은 클래스를 나타내며, 화살표는 온톨로지의 subClassOf 관계를 나타내는데, 화살표로 링크된 부분은 상위클래스를 가리킨다.

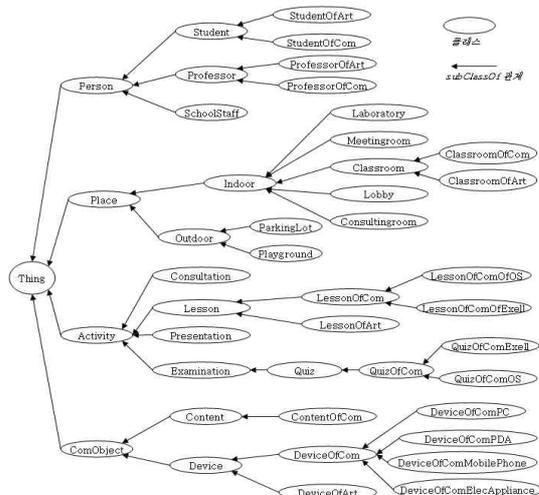


그림 4. CULA-ONT의 클래스 분류  
Fig. 4. Class Hierarchy of CULA-ONT

클래스들의 일례로서, 클래스 Person은 학교에서 활동하는 사람에 대해 일반적인 특징들을 정의한 개념으로 Student, Professor, SchoolStaff 등의 하위클래스를 갖는데, 이것들은 각각 학생, 교수, 교직원을 의미한다. 이 하위클래스들은 서로 형제 클래스이며 disjoint 관계를 가지는데, 이것은 한 구성원이 동시에 둘 이상의 클래스에 속할 수 없음을 명시한 것이다. 그림 5는 정의한 Person의 일부로서, personName은 사람의 이름을 정의하는 데이터 속성이고 이 속성을 통해 스트링 형식의 값을 갖도록 정의한다. 또한 객체 속성 isPresenceIn은 사람의 존재에 대한 제약 조건을 수량 제약으로 정의한 것으로, 이 속성을 통해 관계되는 클래스 Place의 구성원은 오직 1이어야 함을 나타내고 있다. 즉 Person의 한 구성원은 오직 하나의 장소에만 존재해야 함을 표현한 것이다.

```
<owl:Class rdf:about="#Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#personName"/>
      <owl:someValuesFrom rdf:resource="&xsd:string"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#isPresenceIn"/>
      <owl:onClass rdf:resource="#Place"/>
      <owl:qualifiedCardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

그림 5. 클래스 Person의 정의 부분  
Fig. 5. A Partial Definition of Class Person

2. 속성 정의

CULA-ONT 상황 모델에서 두 구성원들 간의 관계를 정의하기 위해 객체 속성과 데이터 속성을 사용한다. 객체 속성은 클래스 구성원들 간의 관계를 기술하고, 데이터 속성은 클래스가 취해야 하는 데이터의 값과 형식을 기술한 것이다. 또한 inverseOf, functional 등 속성의 특징과 속성을 제약하는 수량 제약, 기수 제약, hasValue 제약 등을 사용한다.

표 1은 최상위 클래스와 관련된 5개의 주요 객체 속성들을 정리한 것으로, 각각의 객체 속성들은 대응하는 inverse 속성들을 가지며 도메인의 구성원을 치역의 구성원으로 링크시키는 이진 관계를 가진다. 예를 들어, 객체 속성 hasLocation은 클래스 Place의 한 구성원을 클래스 ComObject의 한 구성원으로 링크시킨다. 객체 속성들은 하위 속성들을 가질 수 있는데, 하위 속성은 구성원들의 좀 더 구체적인 관계를 정의하는데 사용된다. 이러한 하위 속성과 이를 포함하는 상위 속성의 관계를 이용해 새로운 상황 정보를 추론할 수 있다.

표 1. 최상위 클래스와 관련된 객체속성  
Table 1. Object Properties for Top-level Class

객체속성	도메인	치역	inverse 속성
	의미		
hasEvent	Place	Activity	isEventIn
	어떤 장소에 어떤 활동이 있다.		
hasHold	Person	ComObject	isHoldBy
	어떤 사람은 어떤 컴퓨팅 객체를 점유한다.		
hasLocation	Place	ComObject	isLocationIn
	어떤 장소에 어떤 컴퓨팅 개체가 있다.		
hasParticipant	Activity	Person	isParticipantIn
	어떤 활동은 어떤 사람의 참여를 포함한다.		
hasPresence	Place	Person	isPresenceIn
	어떤 장소는 어떤 사람의 존재를 가진다.		

데이터 속성은 하나의 구성원을 XML 스키마 데이터타입 값이나 rdf literal로 링크시킨다. 즉 하나의 구성원과 데이터 값 간의 관계를 기술한다. 표 2는 구성원들이 가질 수 있는 데이터 속성 몇 가지를 소개한 것으로, 도메인은 해당 속성을 가지는 구성원들의 클래스이며, 치역은 구성원이 속성을 통해 가질 수 있는 데이터 타입이다.

표 2. 데이터 속성들의 도메인과 치역  
Table 2. Domain and Range for Data Properties

데이터 속성	도메인	치역	의미
activityEndTime	Activity	&xsd:time	활동 종료 시간
comObjectID	ComObject	&xsd:integer	컴퓨팅 객체 식별자
deviceNetworkng	Device	&xsd:string	디바이스의 네트워크링 가능 여부
personRFIDTagNr	Person	&xsd:string	사람이 소지한 RFID 태그 번호
placeDoorStatus	Place	&xsd:string	문의 개폐 상태

3. 구성원 정의

구성원은 클래스의 인스턴스로 유비쿼터스 학습공간에 실재하는 개체들을 나타낸다. CULA-ONT에 정의된 구성원들은 사람, 장소, 활동 및 컴퓨팅객체들에 대한 인스턴스로, 표 3은 최상위 클래스 타입에 따라 구분한 구성원들의 일부이다.

표 3. 최상위 클래스와 관련된 구성원들의 예  
Table 3. Some of Individuals for Top-level Class

클래스	구성원	의미
Activity	QuizOfComOS-1	퀴즈 풀이 활동
ComObject	DeviceOfComElecAppliance-Projector-422	빔 프로젝터
Person	StudentOfCom-001	학생
Place	ClassroomOfCom-422	컴퓨터학과 교실

4. CULA-ONT 상황 모델의 일치성 검사

OWL-DL을 사용해 기술된 CULA-ONT 상황 모델은 추론기에 의해 처리될 수 있다. 본 논문에서는 상황 모델을 설계하는 과정에서 FaCT 추론기를 사용하였는데, 추론기는 상황 모델의 일치성을 테스트하고, 어떤 클래스가 다른 클래스의 하위클래스인지 테스트 하는데 사용된다. 예를 들어 그림 5의 클래스 Person의 정의를 보자. 이 클래스는 사람의 존재에 대한 제약 조건으로, 객체 속성 isPresenceIn을 통해 클래스 Place에 링크되는 Place의 구성원은 오직 1이어야 함을 나타내고 있다. 만약 상황 모델을 정의하는 과정에서 그림 6과 같이 isPresenceIn을 통해 링크되는 구성원을 둘로 정의했다면, 추론기는 일치되지 않는 온톨로지가 있음을 그림 7과 같이 알린다. 따라서 추론기를 사용함으로써 상황 모델의 구축과 유지 관리에 논리적 타당성이 보장될 수 있다.

```

<ProfessorOfCom rdf:about="#ProfessorOfCom612">
  <rdf:type rdf:resource="&owl:Thing"/>
  <isPresenceIn rdf:resource="#Classroom423"/>
  <isPresenceIn rdf:resource="#Classroom422"/>
</ProfessorOfCom>
    
```

그림 6. 사람 존재에 대한 잘못된 온톨로지 표현  
Fig. 6. Wrong Expression of Ontology for Person Presence

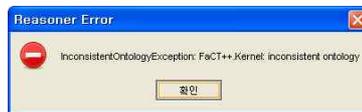


그림 7. 추론기에 의한 온톨로지 일치성 검사  
Fig. 7. Ontology Consistency Test using Reasoner

### IV. 상황 인식 학습 서비스 실험 결과

본 논문은 제안한 유비쿼터스 학습 공간과 CULA 온톨로지 상황 모델을 기반으로 상황 인식 서비스를 실험하기 위해 상황 인식 학습 시스템, 에이전트 및 학습 지원 응용들을 구현하였다. 상황 인식 학습 시스템에서 상황 수집 모듈과 서비스 조정 모듈은 자바 J2SE 1.6을 사용하여 구현하였고, 상황 정보 질의나 갱신 부분은 자바 프레임워크인 Jena(HP Semantic Web Toolkit)[17]를 사용하였다. 상황 인식 학습 시스템과 상황 정보를 주고받는 센서 에이전트, RFID 에이전트 및 기타 에이전트 등은 Visual C++와 Visual Basic으로 구현하였고, 임베디드 게이트웨이에 적재시킨 제어 응용은 C 언어를 사용하여 구현하였다. 사용자에게 최종 제공되는 상황 인식 서비스들은 상황 인식 학습 시스템에 의해 요청되는 학습 지원 응용들로서 Visual C++와 Visual Basic 등으로 구현하였다.

상황 인식 처리를 위해 CULA-ONT 상황 모델의 온톨로지지는 일차 술어 논리 형식을 통해 상황 정보로 표현한다. 일차 술어 논리의 술어 기호는 대상의 특성이나 대상들 간의 관계 기술에 사용되는 것으로, 영문자로 시작하는 문자열을 사용한다. 상황 정보는 (subject, predicate, value)의 형식을 가지며, 참이나 거짓 값을 갖는다. 여기에서 subject는 사람, 장소, 컴퓨팅 객체, 또는 활동 등 상황의 주체를 의미하며, predicate는 personName, isLocationIn, isHoldBy, hasEvent 등 상황 주체가 갖는 특성을 나타내며, value는 predicate를 통해 관련된 다른 상황의 주체나 가질 수 있는 값을 의미한다. 예로, (ProfessorOfCom-11, isPresenceIn, LabProfOfCom-11-408)로 표현된 상황 정보는 교수 ProfessorOfCom-11이 하나의 장소 LabProfOfCom-11-408에 위치해 있음을 의미하는 것이다. 이러한 상황 정보의 표현은 그림 8과 같이 OWL-DL 언어로 작성된 온톨로지 표현을 가진다.

```
<owl:Thing rdf:about="#ProfessorOfCom-11">
  <rdf:type rdf:resource="#ProfessorOfCom"/>
  <isPresenceIn rdf:resource="#LabProfOfCom-11-408"/>
</owl:Thing>
```

그림 8. 상황 정보에 대한 온톨로지 표현 예  
Fig. 8. Ontology Expression for Context Information

#### 1. CULA-ONT 상황 모델 적재

CULA-ONT 상황 모델은 상황 지식을 유지하고 공유하기 위해 상황 지식 베이스에 적재시켰다[17]. 그림 9는 Jena가 제

공하는 패키지를 이용하여 상황 모델에 대한 OWL 파일을 상황 지식 베이스로 적재시키는 부분이다.

```
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.rdf.model.ModelMaker;
public class Main {
  public static final String ONT1 = "CULA-ONT";
  . . .
  ModelMaker maker=po.getRDBMaker( s_dbURL,s_dbUser,
                                  s_dbPw, s_dbType, false );
  po.listClasses( maker, s_source );
}
private static void processArgs( String[] args ) { . . . }
private static String getDefaultSource() {
  OntDocumentManager.getInstance().addAltEntry(
    ONT, "file:./ontologies/cula-ont/cula-ont.owl" );
return ONT; }
}
```

그림 9. CULA-ONT의 OWL 파일을 적재하는 코드 부분  
Fig. 9. A Partial Code for Loading CULA-ONT OWL File

#### 2. 상황 인식 학습 시스템

상황 인식 학습 시스템은 CULA-ONT 상황 모델을 기반으로 상황 정보를 관리하며, 센서나 에이전트들로부터 동적인 상황 정보를 수집하고, 상황 정보를 바탕으로 새로운 상황 정보를 추론한다. 또한 새로운 상황 정보에 따라 사용자에게 제공할 서비스를 요청한다. 그림 10은 상황 인식 학습 시스템이 에이전트들로부터 동적인 정보를 수신하는 시점부터 상황 인식 서비스를 요청하기까지의 상황 처리 과정을 도식화한 것이다.

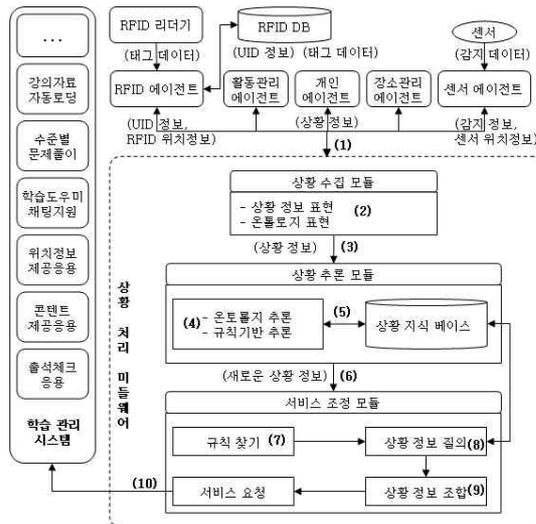


그림 10. 상황 인식 서비스를 위한 처리 과정  
Fig. 10. Process for Context Aware Service

처리 과정을 보면, 에이전트들은 새로운 정보를 상황 수집 모듈로 제공한다(1). 상황 수집 모듈은 수신한 정보를 상황 정보와 온톨로지로 표현하고(2), 상황 추론 모듈에 전달한다(3). 상황 추론 모듈은 온톨로지 추론과 규칙 기반 추론을 통해 새로운 상황 정보를 생성하여(4), 상황 지식 베이스를 갱신하고 일관성을 검사한다(5). 새로운 상황 정보가 서비스 조정 모듈에 제공되면(6), 서비스 조정 모듈은 이 상황 정보에 매칭 되는 상황 정보 구문을 포함한 서비스 규칙을 찾고(7), 서비스 규칙에 지정된 상황 정보들의 구문을 상황 지식 베이스에 질의하고(8), 질의한 결과의 상황 정보들을 조합하여(9), 조합한 결과가 참이 되면 지정된 서비스를 요청한다(10).

2.1 동적인 상황 정보 제공

센서 에이전트, 장소 관리 에이전트 및 RFID 에이전트는 임베디드 게이트웨이 및 RFID 리더기로부터 동적인 데이터를 획득하여 이를 상황 인식 학습 시스템의 상황 수집 모듈에 제공하게 된다. 임베디드 게이트웨이의 제어 응용은 지정된 장소의 환경적 정보 즉 온도와 조도의 감지한 값을 센서 에이전트에게 전달하고, 보드의 릴레이에 연결된 문의 개폐, 전등, 에어컨, PC 등의 온오프 정보를 장소 관리 에이전트에게 전달한다. 제어 응용은 임베디드 게이트웨이의 마이크로프로세서에 적재해 두었는데, C 언어를 사용해 구현하였다. 프로토콜 구조는 소켓 타입 UDP, 소켓 번호 5001로 정의하였으며, 명령 패킷은 커맨드와 데이터 영역으로 구성되며, 응답 패킷은 커맨드와 반환 값 영역으로 구성된다. 표 4에 프로토콜을 위한 패킷 구조와 명령어 일부를 나열하였다.

표 4. 프로토콜 구조 및 명령어  
Table 4. Structure of Protocol and Command

명령 패킷 구조		Command	DATA
응답 패킷 구조		Command	Return
명령어	데이터	Return	설명
FIND_IP_		192.168.0.1	IP 가져옴
SET_GATE	192.168.0.1	같음	게이트웨이 IP 세팅
RELAY_ON	1	같음	1번 릴레이 ON
GET_TEMP		26	온도를 가져옴
GET_LIGH		500	조도를 가져옴

에이전트들은 감지된 정보를 상황 수집 모듈로 제공하는데, 그림 11은 센서 에이전트가 상황 인식 학습 시스템의 상황 수집 모듈과 서로 통신하기 위해 소켓을 설정하는 코드의 일부이다.

```
Private Sub cmdConnect_Click()
    If Winssock.State = sckClosed Then
        Label1.Caption = ""
        Winssock.Protocol = sckTCPProtocol
        Winssock.Connect txtHOSTIP.Text, "1234"
        Label1.Caption = "JAVA와의 연결이 성공되었습니다."
    End If
End Sub
```

그림 11. 센서 에이전트 코드의 일부  
Fig. 11. A Partial Code of Sensor Agent

2.2 상황 수집 모듈의 온톨로지 표현

상황 수집 모듈은 센서 에이전트나 RFID 에이전트로부터 감지된 데이터를 얻고, 이를 일차 술어 논리 형식의 상황 정보와 온톨로지로 표현하여 상황 추론 모듈에 전달한다.

그림 12의 왼쪽은 어떤 사람이 그의 RFID 태그를 실습실 문에 있는 RFID 리더기에 대고 들어갈 때, 그 실습실에 사람이 있다는 사실을 인식하는 과정을 설명하며, 오른쪽은 상황 수집 모듈에 의한 상황 정보와 온톨로지 표현 결과이다. RFID 에이전트는 실습실 ClassroomOfCom-423에 있는 리더기로부터 태그 값 "EBDC280F014E0"을 받으면, RFID DB에서 태그 값에 해당하는 사용자의 UID ProfessorOfCom-46을 추출한다. 그리고 사용자의 UID 정보 및 위치 정보를 스트링 형식으로 상황 수집 모듈에 전송한다. 상황 수집 모듈은 사람이 실습실에 존재한다는 속성 hasPresence를 적용해 일차 술어 논리 형식의 상황 정보로 표현하고 상황 정보를 온톨로지 표현으로 변환하여 상황 추론 모듈로 보낸다.



그림 12. 사람 존재의 상황 인식 과정과 온톨로지 표현  
Fig. 12. Process of Context aware for Person Presence and Ontology Expression

2.3 상황 추론 모듈에 의한 새로운 상황 정보 생성

상황 추론 모듈은 상황 정보를 해석하여 새로운 상황 정보를 추론하는 기능으로 온톨로지 추론과 규칙 기반 추론을 도입한다. 적용한 온톨로지 추론은 온톨로지 자체가 가진 추론 기능인 subClassOf, inverseOf 및 FunctionalProperty 등을 사용하는데, 이를 일차 술어 논리로 표현하면 아래와 같다.

- *subClassOf*: (?A rdfs:subClassOf ?B), (?B rdfs:subClassOf ?C)  
⇒ (?A rdfs:subClassOf ?C)
- *inverseOf*: (?P owl:inverseOf ?Q), (?A ?P ?B) ⇒ (?B ?Q ?A)
- *FunctionalProperty*: (?P rdf:type owl:FunctionalProperty), (?A ?P ?B),  
(?A ?P ?C) ⇒ (?B = ?C)

규칙 기반 추론은 온톨로지 추론에 의해 얻을 수 없는 상황을 미리 정한 추론 규칙에 따라 여러 가지 상황 정보로부터 새로운 상황을 추론하는 것이다. 추론 규칙은 여러 가지 상황들을 일차 술어 논리에 맞게 조합한 상황들의 집합으로 볼 수 있는데, 이 규칙은 몇 가지 상황 정보를 부울 대수로 조합하여 그 결과가 참인 경우 새로운 상황 정보를 정의하는 것이다. 그림 13은 규칙 기반 추론의 예로서, 임의의 한 사람이 임의의 컴퓨팅 객체를 사용하고 있고, 그 컴퓨팅 객체가 임의의 장소에 있다면, 그 사람 또한 그 장소에 있다고 추론하는 것이다. 속성 *hasHoldOfStaticDevice*는 *hasHold*의 하위 속성으로서 클래스 *Person*의 한 구성원을 클래스 *ComObject*의 한 구성원으로 링크시키는데, 네트워크를 통한 원격 접속이 아니라 직접 사용 중임을 의미하는 속성이다. 추론 규칙에서, 기호  $\Rightarrow$ 의 왼쪽에 있는 두 개의 상황 정보는 각각 온톨로지 1과 2의 온톨로지 표현과 대응되며, 기호  $\Rightarrow$ 의 오른쪽에 있는 상황 정보는 온톨로지 3과 대응된다.

추론 규칙

(?Person, *hasHoldOfStaticDevice*, ?ComObject), (?ComObject, *isLocationIn*, ?Place) ⇒ (?Person, *isPresenceIn*, ?Place)

온톨로지 1

```
<owl:Thing rdf:about="#StudentOfArt-002">
  <hasHoldOfStaticDevice rdf:resource="#DeviceOfArtPC-1"/>
</owl:Thing>
```

온톨로지 2

```
<owl:Thing rdf:about="#DeviceOfArtPC-1">
  <isLocationIn rdf:resource="#Meetingroom-25"/>
</owl:Thing>
```

온톨로지 3

```
<owl:Thing rdf:about="#StudentOfArt-002">
  <hasHoldOfStaticDevice rdf:resource="#DeviceOfArtPC-1">
  <isPresenceIn rdf:resource="#Meetingroom-25"/>
</owl:Thing>
```

그림 13. 규칙 기반 추론의 예

Fig. 13. Example of Rule-based Reasoning

2.4 서비스 조정 모듈에 의한 상황 인식 서비스 제공

서비스 조정 모듈은 현재의 변화된 상황에 따라 서비스 규칙에 지정된 구문의 상황 정보들을 얻고 이것들을 조합하여 결과가 참인지 검사하여 그 결과가 참인 경우 지정된 서비스

를 수행시킨다. 서비스 조정 모듈에서 서비스를 요청하기 위해 상황 정보를 처리하는 과정은 다음 단계와 같다.

- 단계 (1) 서비스 조정 모듈로 새로운 상황 정보가 전송된다.
- 단계 (2) 서비스 규칙들의 첫 번째 상황 정보 구문을 검사하여 새로운 상황 정보와 매칭된 구문을 포함한 서비스 규칙을 찾는다.
- 단계 (3) 찾은 서비스 규칙의 나머지 구문들에 해당하는 상황 정보를 차례대로 상황 지식 베이스로 질의하여 상황 정보를 수집한다.
- 단계 (4) 수집된 상황 정보들을 조합한다.
- 단계 (5) 결과가 참이면, 규칙에 지정된 서비스를 요청한다.

서비스 조정 모듈을 통해 자동으로 제공할 수 있는 상황 인식 서비스로 문 열기, 에어컨 동작, 출석체크, 수업준비, 퀴즈풀이 학습도우미 채팅지원, 수업 후 전자제품 끄기 서비스 등을 구현하였다. 대표적으로 문 열기와 퀴즈 풀이 학습 도우미 채팅 지원 서비스를 소개한다.

(1) 문 열기 서비스

이 서비스는 활동이 예정된 장소에 정당한 사용자가 최초 입장할 때 그 장소의 문을 열고 전등을 켜는 서비스이다. 수업 *LessonOfComOS-1*이 예정된 실습실 *ClassroomOfCom-422*에 사용자 *StudentOfCom-001*이 자신의 RFID 태그를 대고 최초 입장하는 상황을 실험하여, 실습실 문을 자동으로 여는 결과를 확인하였다. 이 서비스는 한 번만 수행하면 되기 때문에, 관련 규칙에 플래그 항목을 두고 사용 여부를 설정한다. 문 열기 서비스를 제공하기 위해 서비스 조정 모듈에서 수행하는 상황 정보 처리 과정은 표 5와 같다.

단계 (1)에서 상황 정보 (*ClassroomOfCom-422*, *hasPresence*, *StudentOfCom-001*)를 수신하면, 단계 (2)는 규칙들 중 첫 번째 구문에 속성 *hasPresence*를 포함한 서비스 규칙을 찾는다. 상황 정보와 구문을 비교하여 ?Place에 해당하는 구성원은 *ClassroomOfCom-422*이며, ?Person에 해당하는 구성원은 *StudentOfCom-001*임을 얻게 된다. 단계 (3)은 단계 (2)에서 찾은 구성원을 적용해 구문 2의 질의는 구성원 *ClassroomOfCom-422*가 속성 *hasEvent*를 통해 링크하고 있는 구성원을 찾는 형식이 되고, 반환된 값 *LessonOfComOS-1*은 ?Activity에 할당한다. 다음, 구문 3의 질의는 구성원 *StudentOfCom-001*로 관계하는지 질의한다. 단계 (4)는 수집된 상황 정보들을 조합하는데, 문 열기 서비스 규칙의 경우 논리곱( $\wedge$ )으로 조합되어 있다. 단계 (5)는 조합 결과가 참일 경우, 해당 서비스 *onGate(ClassroomOfCom-422)*를 요청한다.

표 5. 서비스 조정 모듈의 처리 과정-1  
Table 5. Process-1 of Service Coordination Module

서비스 조정 모듈	
단계 (1) 새로운 상황 정보 수신	
상황정보	(ClassroomOfCom-422, hasPresence, StudentOfCom-001)
단계 (2) 규칙 찾기	
매칭 구문	(ClassroomOfCom-422, hasPresence, StudentOfCom-001) ( ?Place, hasPresence, ?Person )
찾은 규칙	(?Place, hasPresence, ?Person) ^ (?Place, hasEvent, ?Activity) ^ (?Activity, ?Property, ?Person) ⇒ onLight(?Place), onGate(?Place)
단계 (3) 상황 정보 수집	
구문1	결과 (ClassroomOfCom-422, hasPresence, StudentOfCom-001)
구문2	구문 (?Place, hasEvent, ?Activity)
	질의 (ClassroomOfCom-422, hasEvent, ?Activity) ClassroomOfCom-422에 있는 활동을 찾아라.
구문3	결과 (ClassroomOfCom-422, LessonOfComOS-1)
	구문 (?Activity, ?Property, ?Person)
	질의 (LessonOfComOS-1, ?Property, StudentOfCom-001) LessonOfComOS-1이 StudentOfCom-001로 링크되는 속성을 찾아라.
결과	(LessonOfComOS-1, hasLearner, StudentOfCom-001)
단계 (4) 상황 정보 조합	
	(ClassroomOfCom-422, hasPresence, StudentOfCom-001) ^ (ClassroomOfCom-422, hasEvent, LessonOfComOS-1) ^ (LessonOfComOS-1, hasLearner, StudentOfCom-001)
단계 (5) 삽입, 서비스 요청	
서비스	onLight(ClassroomOfCom-422), onGate(ClassroomOfCom-422)

단계 (3)에서 상황 지식 베이스로의 질의는 Jena의 RDQL(RDF Data Query Language)을 사용하는데, 그림 14의 (a)는 구문 2에 대한 RDQL의 질의 유형이며, (b)는 서비스 조정 모듈에서 질의를 수행하는 코드의 일부이다.

```

SELECT ?Activity WHERE
(ClassroomOfCom-422, hasEvent, ?Activity)
(a) 문 열기 서비스 규칙의 구문 2에 대한 RDQL 질의 유형
String queryString1 =
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> "+
"PREFIX Ontology22:
<http://cit.wst.ac.kr/ontologies/2009/9/Ontology22.owl#> "+
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "+
".
"SELECT ?Activity " +
"WHERE(Ontology22:ClassroomOfCom-422,
Ontology22:hasEvent, ?Activity );
Query query = QueryFactory.create(query_string1);
QueryExecution qe = QueryExecutionFactory.create(query, infmodel);
ResultSet results = qe.execSelect();
(b) Jena가 지원하는 RDQL을 이용한 질의 코드의 일부
    
```

그림 14. 상황 정보 질의의 유형과 관련 코드  
Fig. 14. Context Query Type and Related Code

onLight()와 onGate()는 ClassroomOfCom-422의 장소 관리 에이전트에게 전등을 켜고 문을 열어 달라는 메시지를 전송한다. onLight() 경우, 장소 관리 에이전트는 전등을 켜달라는 메시지에 해당하는 "RELAY\_ON11"을 스트링 형식으로 임베디드 게이트웨이의 제어 응용에 전달하고, 제어 응용은 해당 릴레이에 연결한 전원을 온 시켜 전등을 켤다. 그림 15는 전자제품 온오프를 제어하는 장소 관리 에이전트 코드의 일부로서 "RELAY\_ON11"로 정의하여 임베디드 게이트웨이의 제어 응용으로 이벤트를 전송하는 코드이다.

```

Private Sub Cmd_Output_Click()
Dim Str_SendBuf As String
If Cmd_Start.Enabled = False Then
Winsock1.RemoteHost = Txt_IP.Text
Winsock1.RemotePort = 50001
Str_SendBuf = "RELAY_ON11"
Winsock1.SendData Str_SendBuf
End If
End Sub
    
```

그림 15. 전등을 켜는 코드  
Fig. 15. A Partial Code for Turn-on the Light

그림 16은 센서 및 장소 관리 에이전트 응용의 제어 화면으로 서비스 조정 모듈로부터 전등을 켜 달라는 요청을 받고 해당 서비스를 수행한 결과를 보여주고 있다.

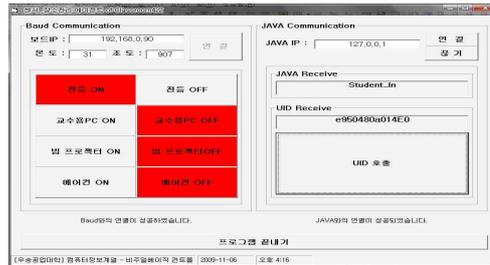


그림 16. 센서 및 장소 관리 에이전트에 대한 제어 화면  
Fig. 16. Control View of Sensor and Place Management Agent

(2) 퀴즈 풀이 및 학습 도우미 채팅 지원 서비스  
이 서비스는 학생들이 참여한 퀴즈 풀이 결과에 따라 성적이 낮은 학생을 선정하고 이 학생이 퀴즈 주제와 관련된 학습 도우미에게 도움을 받을 수 있도록 채팅 환경을 제공하는 서비스이다. 서비스 조정 모듈에서의 처리과정은 표 6과 같다. 활동 에이전트의 하나인 퀴즈 에이전트는 참여한 학생 정보를 상황 인식 학습 시스템으로 전송하고, 상황 인식 학습 시스템은 전송된 정보를 상황 정보로 갱신한다. 퀴즈 에

이전트는 학생들이 작성한 퀴즈의 답안을 채점해, 채점 결과에 따라 성적이 낮은 학생 정보를 상황 수집 모듈로 전송한다. 상황 수집 모듈은 상황 정보 (QuizOfComOS-1, hasQuizParticipant, StudentOfCom-002)를 서비스조정 모듈로 보낸다. 서비스 조정 모듈은 퀴즈 활동에 링크된 학습 도우미를 찾고, 그가 사용 중인 디바이스의 IP 주소를 퀴즈 에이전트에게 전송한다.

표 6. 서비스 조정 모듈의 처리 과정-2  
Table 6. Process-2 of Service Coordination Module

서비스 조정 모듈	
단계 (1) 새로운 상황 정보 수신	
상황정보	{QuizOfComOS-1, hasQuizParticipant, StudentOfCom-002}
단계 (2) 규칙 찾기	
매칭구문	{QuizOfComOS-1, hasQuizParticipant, StudentOfCom-002} ( ?Activity, hasQuizParticipant, ?Person1 )
찾은규칙	{?Activity, hasQuizParticipant, ?Person1} ^ {?Activity, hasSupporter, ?Person2} ^ {?Person2, hasHold, ?Device} ^ {?Device, deviceNetworking, "Enable"} ^ {?Device, deviceIP, ?value} => sendMessage(QuizAgent, ?value)
단계 (3) 상황 정보 수집	
구문1	결과 {QuizOfComOS-1, hasQuizParticipant, StudentOfCom002}
구문2	구문 {?Activity, hasSupporter, ?Person2}
	질의 {QuizOfComOS-1, hasSupporter, ?Person2} 퀴즈 QuizOfComOS-1의 학습 도우미를 찾아라.
구문3	결과 {QuizOfComOS-1, hasSupporter, StudentOfCom-201}
	구문 {?Person2, hasHold, ?Device}
구문4	질의 {StudentOfCom-201, hasHold, ?Device} StudentOfCom-201이 사용 중인 Device를 찾아라.
	결과 {StudentOfCom-201, hasHold, DeviceOfComPC-021}
구문5	구문 {?Device, deviceNetworking, "Enable"}
	질의 {DeviceOfComPC-021, deviceNetworking, "Enable"} DeviceOfComPC-021이 네트워킹이 가능한가?
구문6	결과 {DeviceOfComPC-021, deviceNetworking, "Enable"}
	구문 {?Device, deviceIP, ?value}
구문7	질의 {DeviceOfComPC-021, deviceIP, ?value} DeviceOfComPC-021의 IP 주소를 찾아라.
	결과 {DeviceOfComPC-021, deviceIP, "168.192.68.54"}
단계 (4) 상황 정보 조합	{QuizOfComOS-1, hasQuizParticipant, StudentOfCom-002} ^ {QuizOfComOS-1, hasSupporter, StudentOfCom-201} ^ {StudentOfCom-201, hasHold, DeviceOfComPC-021} ^ {DeviceOfComPC-021, deviceNetworking, "Enable"} ^ {DeviceOfComPC-021, deviceIP, "168.192.68.54"}
단계 (5) 참이면, 서비스 요청	
서비스	sendMessage(QuizAgent, "168.192.68.54")

그림 17은 퀴즈 에이전트의 실행 화면으로 학생들에게 문제를 제공하고 답을 채점하여 그 결과를 전송할 수 있다. 또한 학습 도우미를 채팅방에 초청하여 학생과 대화할 수 있게 한다.



그림 17. 퀴즈 및 채팅 지원 응용의 실행 화면  
Fig. 17. View of Application for Quiz and Chatting

## V. 결론

상황 인식 기반의 유비쿼터스 학습 서비스를 지원하기 위해, 기존 LAN 환경에서 이루어지는 학교 네트워크를 대상으로 USN을 활용한 유비쿼터스 학습 공간을 제안하고, OWL을 사용해 유비쿼터스 학교 환경에 대한 온톨로지 기반의 CULA-ONT 상황 모델을 구축하였다. 또한 이를 기반으로 한 상황 인식 학습 시스템을 구현하여 상황 인식 서비스의 사례들을 실험함으로써 제안한 유비쿼터스 학습 공간과 온톨로지 상황 모델의 적합성을 제시하였다.

본 논문의 장점으로, 실제 학교를 대상으로 온톨로지 상황 모델을 이용해 상황 인식 학습 서비스를 제공하는 시스템에 대한 연구가 거의 없는 상황에서, 무무선 LAN 환경의 기존 학교 공간을 대상으로 하였고, 손쉽게 구할 수 있는 센서와 RFID 등의 유비쿼터스 기기를 활용한 USN 기반의 네트워크를 구성함으로써 유비쿼터스 학교 망 구축에 현실적인 접근 방안을 제시하였다. 또한 자체 제작한 임베디드 게이트웨이를 상황 정보가 많이 발생하는 장소에 배치하여 별도의 USN 환경을 구축함으로써 센서 네트워크 관리의 효율성을 제고할 수 있는 방법을 제시하였다. 특히 상황 모델을 구축함에 있어, 기기가 처리할 수 있는 정보 공간을 표현하는 온톨로지를 도입함으로써, 유비쿼터스 환경에 있는 기기종의 다양한 기기들 및 서비스들 간에 상황 공유와 상황 추론 및 상황 모델의 재사용을 가능하게 할 수 있다는 장점을 가진다. 비록 본 논문이 제시한 온톨로지 상황 모델이 학교에 대한 모든 객체들과 상황들을 표현하지는 못하지만, 온톨로지의 재사용 가능성을 고려해볼 때 수정과 확장이 가능하리라 사료된다. 상황 인식 서비스의 제공에 대해 모바일 기기의 지원을 아직 구현하지 못한 점과 대용량의 상황 정보를 수용하기 위한 자체의 데이터베이스를 구축하지 못한 점은 향후 계속해서 보완하고 연구해야 할 과제이다.

## 참고문헌

- [1] ihcho, "Research on Developing a Next Generation Learning Model in a Ubiquitous Computing Environment," KERIS, CR 2006-4, 2006.
- [2] jhseo, "Development of teaching and learning model in ubiquitous Learning Environment," KERIS, CR 2005-12, 2005.
- [3] shkwuan, "Research on u-learning Korea Road Map," KERIS, CR 2006-11, 2006.
- [4] iwpark, "Configuration Plan of Future Classroom for Ubiquitous Environment," KERIS, CR 2006-14, 2006.
- [5] yjlee, "A Study on Development of Ubiquitous Learning Device," KERIS, CR 2006-19, 2006.
- [6] knkim, Research on u-Learning Supporting System (u-LSS) and Development of Prototype, KERIS, KR 2007-14, 2007.
- [7] jhlim, Construction of u-Campus for University, KERIS, RM 2006-63, 2006.
- [8] KERIS, Evolution of School Network in Ubiquitous Learning Environment, RM 2006-99, 2006.
- [9] S. Yang, "Context Aware Ubiquitous Learning Environments for Peer-to-Peer Collaborative Learning," Educational Technology & Society, 9 (1), pp. 188-201, 2006.
- [10] jhpark and jhkang, "An Intelligent Service Middleware Using Ontology and Rule in Ubiquitous Sensor Network Environments," Journal of KSCI, Vol. 15, No. 10, 2010.
- [11] jhbaek, "The Design and Implementation of Intelligent Internet Outlet for Real-Time Scheduling Control," Journal of KSCI, Vol. 15, No. 10, 2010.
- [12] A. Dey, G. Abowd, and D. Salber, "A Context-Based Infrastructure for Smart Environments," Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99), pp. 14-128, 1999.
- [13] A. Dey and G. Abowd, "Towards Better Understanding of Context and Context-Awareness," Proceedings of the CHI 2000 Workshop, The Hague, April 1-6, 2000.
- [14] B. Schilit, D. Hilbert, and J. Trevor, "Context-Aware Communication," IEEE Wireless Communication, 2002.
- [15] T. Gruber, "Toward Principals for the Design of Ontologies Used for Knowledge Sharing," Journal of Human-Computer Studies, Vol. 43, 1995.
- [16] E. Christopoulou, C. Goumopoulos, and A. Kameas, "An Ontology-Based Context Management and Reasoning Process for UbiComp Applications," Joint Soc-EUSAI conference, pp. 265-270, Grenoble, October 2005.
- [17] Jena, <http://jena.sourceforge.net/index.html>

## 저자 소개



### 홍명우

1993 : 충남대학교 컴퓨터공학과 공학사.

1995 : 충남대학교 컴퓨터공학과 공학석사.

2010 : 안동대학교 멀티미디어공학과 공학박사

현재 : 우송정보대학 컴퓨터정보계열 교수

관심분야 : 상황인식컴퓨팅, 유비쿼터스  
Email : mwhong@wst.ac.kr



### 이용환

1986 : 충남대학교 계산통계학과 이학사.

1989 : 건국대학교 전자계산학과 공학석사.

1999 : 건국대학교 전자계산학과 공학박사

현재 : 우송정보대학 방송영상디자인과 교수

관심분야 : 영상처리 및 인식, 유비쿼터스  
Email : ywlee@wsi.ac.kr