

상황 온톨로지를 활용한 QoS 제공 상황인식 프레임워크

서동우 · 이재열[†]

전남대학교 산업공학과

Context-Aware Framework Using Context Ontology for Supporting QoS-based Services

Dong Woo Seo · Jae Yeol Lee

Department of Industrial Engineering, Chonnam National University, 300 Yongbong-dong, Buk-gu, Gwangju

In the future, the ubiquitous computing environment will provide users with context-aware services, intelligently interacting with omnipresent resources and surrounding environments at any location and time. Therefore, the ubiquitous computing environment requires context-aware applications in order to gather and analyze context information in various situations. However, existing context-aware applications have mainly focused on providing services and adapting themselves to users based on user related contexts. But this environment requires not only user related contexts but also Quality-of-Service (QoS) related contexts. In this paper we propose a framework for supporting user-oriented QoS in ubiquitous environments. The proposed approach adopts a semantic ontology and a neural network algorithm in order to reason about explicit and uncertain contexts. We also show the possibility of applying the framework to real environments such as collaborative engineering service, automobile maintenance service and ubiquitous home service.

Keyword: context-aware and ubiquitous computing, ontology, jena, neural network, semantic web

1. 서론

최근에 가장 각광을 받고 있는 기술은 유비쿼터스 컴퓨팅(Ubiquitous Computing)이다. 유비쿼터스 컴퓨팅은 언제, 어디서나 사용이 가능한 컴퓨팅 환경을 지칭하는 말로 다양한 종류의 컴퓨터가 사람, 사물, 환경 속에 내재되어 있고, 서로 연결되어 필요한 서비스를 효과적으로 제공할 수 있는 컴퓨팅 환경을 의미한다(Weiser, 1991). 특히, 모바일 기기 및 임베디드 컴퓨팅 기술의 발전으로 일상생활에서의 컴퓨팅 환경은 언제 어디서든지 멀티미디어, 엔지니어링 협업, 유비쿼터스 홈, 텔레메틱스 기반의 자동차 등의 서비스들을 제공할 수 있게 되었다. 또한, 앞으로는 상황인식 컴퓨팅 기술의 발전으로 사용자의 필

요와 의도를 미리 예상하고 사용자의 상황(Context)에 가장 적합한 서비스를 제공할 수 있게 될 것이다(Bisdikian *et al.*, 2002; Chen *et al.*, 2004; Fano and Gershman, 2005; Rhee *et al.*, 2006).

시맨틱 웹에서의 온톨로지(Ontology)는 “개념화의 규정”을 말한다. 즉 해당영역의 개념들과 이들 개념 간의 상호 관계를 정의하는 것을 온톨로지라고 한다. 이러한 온톨로지는 상황인식을 상호 운영하고 처리할 수 있도록 도와준다. 온톨로지를 이용하여 상황을 모델링함으로써 쉬운 추론 및 패턴 분석, 사람과 기계, 기계 사이의 원활한 커뮤니케이션, 상황정보 공유의 용이성 및 확장성, 다양한 응용 서비스 관리의 편리함을 얻을 수 있다(Henricksen *et al.*, 2002).

상황은 장소, 장소의 환경적인 속성, 함께 존재하고 있는 사

본 연구는 BK21 프로그램과 LG 연암문화재단에 의해서 지원되었음.

[†]연락처 : 이재열 교수, 500-757 광주광역시 북구 용봉동 300번지 전남대학교 산업공학과, Fax : 062-530-1789,

E-mail : jaeyeol@chonnam.ac.kr

2007년 05월 접수, 2회 수정 후 2007년 08월 게재확정.

람들, 관련된 디바이스에 대한 정보 등을 의미한다. 상황인식 컴퓨팅은 일반적으로 지능형 로봇, 편리한 생활을 위한 스마트 공간(Smart Space), 텔레매틱스 기반의 자동차와 같은 분야에서 이용되었다. 그러나 기존의 연구들은 사용자 관련 상황인식, QoS를 고려한 상황인식을 충분히 적용하지 못함으로써 사용자의 만족도를 향상시키지 못하였다(Wang *et al.*, 2004).

기존의 몇몇 연구에서는 상황인식 기반의 서비스 인프라를 제공하는 미들웨어 연구를 활발히 진행해 왔다. GATECH에서는 상황정보를 수집, 분석 및 해석, 그리고 응용서비스에 전달하는 Context Toolkit(Dey and Abowd, 2000)을 개발하였다. RSCM Scarlet(Yau *et al.*, 2002)은 이질적인 플랫폼 간에 상황정보를 교환하기 위한 방안을 제시하였다. 하지만 기존의 연구들은 사용자와 관련된 상황인식에 주된 초점을 두었다. 상황인식 프레임워크(Framework)로서 사용자 중심의 서비스를 제공하기 위해서는 사용자와 관련된 상황과 사용자의 서비스와 관련된 QoS도 함께 고려할 때 성취될 수 있다(Alber *et al.*, 2002).

따라서 본 논문에서는 사용자 중심의 QoS를 제공하기 위한 상황인식 컴퓨팅 프레임워크를 제시하고 이를 이용한 응용시스템을 구현하여 이를 검증한다. 이를 위하여 확장형 상황 온톨로지를 정의하고 이를 근간으로 상황을 모델링한다. 모델링된 상황 온톨로지를 이용하여 상황지식을 상호 운용할 수 있는 방법을 제시하며 또한 신경망이 어떻게 사용자와 관련된 QoS 상황추론을 지원해줄 수 있는지를 논의한다. 마지막으로 엔지니어링 협업 서비스, 자동차 정비 서비스, 유비쿼터스 홈 서비스를 바탕으로 하는 응용시스템을 구현하여 제시된 상황인식 프레임워크가 어떻게 사용자 중심의 서비스를 제공할 수 있는지 보여준다.

2. QoS 제공 프레임워크

본 논문의 주된 목적은 유비쿼터스 환경하에서 사용자 중심의

QoS를 제공하는 상황인식 컴퓨팅 프레임워크를 제시하는 것이다. 제시된 프레임워크는 <Figure 1>과 같이 4단계의 계층으로 구성이 되어 있다. ① Interface Layer, ② QoS Service Layer, ③ Context-Aware Layer, ④ Context Layer.

Interface Layer는 Service Agent를 이용하여 여러 자원들로부터 얻어진 하위 상황정보를 Context-Aware Layer에 통보하는 역할을 한다. 자원들은 RFID와 같은 다양한 센서들과 장치들로 구성되어 있다. 특히, 본 연구에서는 CAMUS의 Service Agent(Kim *et al.*, 2005)를 재구성하여 컴포넌트간의 효과적인 서비스 관리와 범용성을 제공하도록 하였다. Service Agent는 Context-Aware Server로부터 얻어진 최종적인 상황을 이용하여 장치제어, 일정관리와 같은 서비스를 제공해 준다.

QoS Service Layer는 사용자의 상황에 따라서 서비스와 관련된 품질요소들을 조율하는 역할을 한다. 서비스와 관련된 품질요소들은 CPU 사용량, 메모리의 사용량과 같은 컴퓨팅 자원이나 네트워크 대역폭 등의 네트워크 상태 그리고 사용자의 선호도와 보안 등의 요소들로 구성되어 있다. 이를 위하여 QoS Service Layer는 Interface Layer와 Context-Aware Layer 중간에 위치하여 Service Agent와 Context-Aware Server간의 통신 채널을 제공해 주고 이를 효과적으로 관리한다. 또한, QoS 상황 정보들을 추론하기 위하여 Context-Aware Server를 이용한다.

Context-Aware Layer는 상황인식 처리를 담당한다. 이 레이어는 Interface Layer와 QoS Service Layer부터 이벤트를 통해 전달된 상황정보, QoS 상황정보를 바탕으로 지식들을 관리, 추론, 검색하는 기능을 제공한다. 특히, 신경망 기술을 보완적인 수단으로 활용하여 단순히 명시적인 온톨로지 추론으로 처리하기 어려운 상황을 처리한다.

본 연구에서는 다양한 센서정보 수집과 상황인식 응용 개발에 필요한 제반 기능을 지원하기 위하여 ETRI에서 개발된 CAMUS를 활용한다(Kim *et al.*, 2005). CAMUS는 환경 내 상황정보를 이용하여 다양한 서비스를 연동하여 보다 능동적인 서비스를 제공하기 위한 시스템 하부엔진으로 활용된다. 하지만, CAMUS

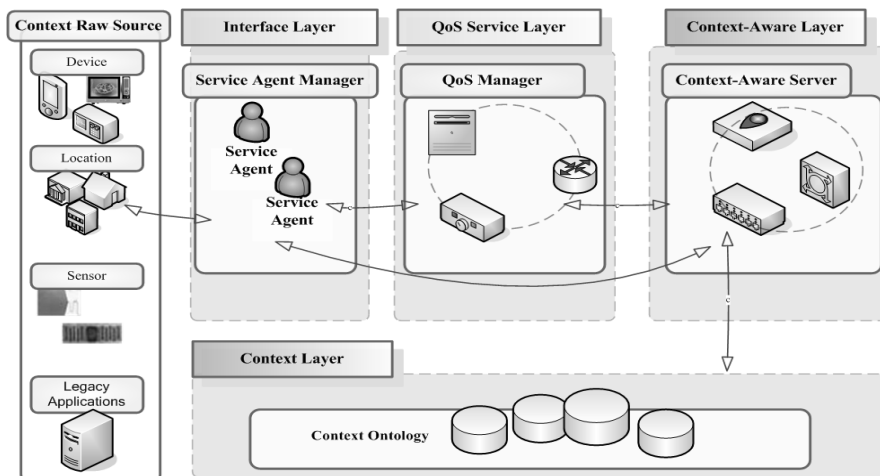


Figure 1. Framework for supporting QoS

는 수집된 상황정보를 효과적으로 통합 및 모델링하고 상호 운용할 수 있는 방법을 지원하지 않는다. 뿐만 아니라, 불확실한 상황과 QoS 상황들을 지원되지 않는다.

Context Layer는 상황정보와 QoS 상황정보를 저장하는 상황 온톨로지 저장소 역할을 한다. 이러한 상황 온톨로지는 CONON (Alber *et al.*, 2002), CC/PP(2004), FOAF(2006) 등의 잘 정의된 상위 온톨로지를 확장하여 모델링하였다.

2.1 Interface Layer의 Service Agent Manager

Service Agent는 다양한 외부 센서나 장치 및 시스템들을 Context-Aware Server와 연동될 수 있도록 프록시 역할을 하는 모듈이다. 즉, 환경내부 센서로부터 획득한 정보를 Context-Aware Server에 전달하고, Context-Aware Server로부터의 최종 명령을 받아 환경 내 장치를 제어, 어플리케이션 실행 등의 다양한 서비스를 제공한다. Service Agent는 Service Agent Manager가 제공하는 실행환경에서 수행되며, Service Agent Manager는 수행 중인 Service Agent들에 대한 서비스 탐색 기능을 제공함으로써 원하는 Service Agent에 접근할 수 있도록 해준다(Kim *et al.*, 2005). <Figure 2>는 Service Agent Manager의 구조를 보여준다.

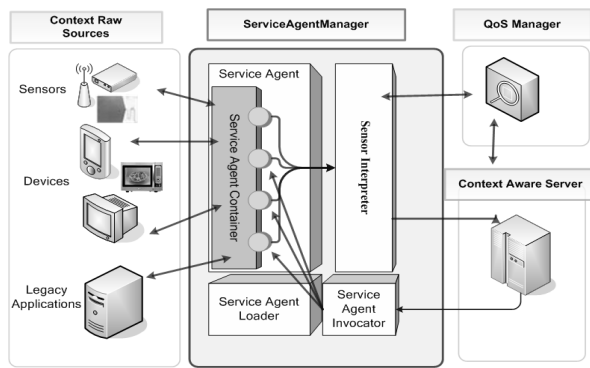


Figure 2. Service Agent Manager

다양한 Service Agent들은 Service Agent Loader에 의해 Service Agent Container에 탑재되어 운용(설치/변경/제거)된다. 이렇게 탑재된 Service Agent들은 Context-Aware Server와 통신하게 된다. 서비스 에이전트로부터 획득되는 정보는 Sensor Interpreter를 거치면서 정제되거나 합성된다. 이러한 과정을 거친 최종 정보는 Context-Aware Server에 전달된다.

Context-Aware Server로부터의 Service Agent 호출은 Context-Aware Server 통신 프레임워크를 통해 직접 호출될 수도 있으며, Service Agent Invocator를 통해 호출될 수 있다. 또한 서비스 이용 도중에 사용자와 관련된 QoS 상황 변화가 발생시 QoS Manager에 의해서 서비스에 대한 품질이 동적으로 조율된다.

2.2 Context-Aware Server

Context-Aware Server는 CAMUS의 Context-Aware Server를 기반

으로 재구성하였다(Kim *et al.*, 2005). Context-Aware Server는 다양한 지식들을 관리, 추론, 검색하는 기능을 제공한다. Context-Aware Server의 Context Monitor는 Service Agent로부터 센서 정보들을 전달받는다. <Figure 3>에서 볼 수 있듯이 서비스 에이전트로부터 얻어진 하위상황정보(사용자ID, 위치, 시간, 날씨 등)는 Context Manager를 통해 상황 온톨로지에 저장된다.

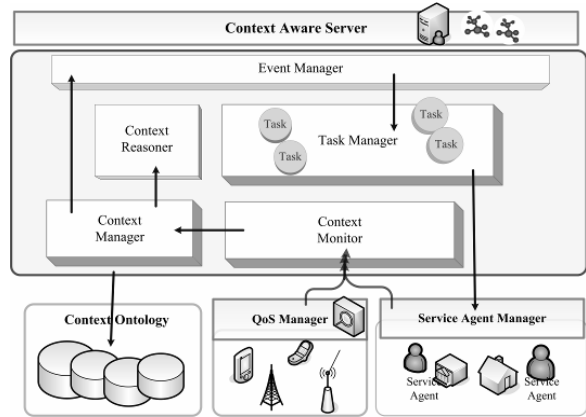


Figure 3. Context-Aware Server

또한, Context Manager는 추후 이벤트에 따른 태스크(Task)의 실행을 위하여 발생된 상황변화를 Event Manager에게 통보하게 된다. Event Manager는 통보된 상황변화를 태스크에서 이용할 수 있는 형태로 변환하여 추후에 적절한 이벤트를 Task Manager에게 통보하는 역할을 한다. Context Reasoner는 저장된 하위 상황정보들을 보다 고수준의 상황정보(의도, 활동 등)로 해석하여 사용자에게 서비스를 제공한다.

위의 과정을 통해서 구축된 상황정보는 각각 Task Manager를 통해 관리된다. 태스크는 Event Manager를 통해 전달되는 이벤트를 온톨로지 규칙에 적용하며 규칙을 만족시키면 명시된 행위를 발생시킨다. 이 행위는 Service Agent에게 집안의 온도를 제어하는 등의 장치제어, 일정관리와 같이 사용자에게 특정한 서비스를 제공하는 형태가 된다.

2.3 QoS 제공을 위한 상황 온톨로지

현실세계의 상황정보를 제공해주는 지식베이스를 구축하기 위해서는 다양한 상황정보를 고려할 필요가 있다. 본 논문에서는 세 가지 범주의 상황클래스를 정의하였다. ①일반적인 도메인 개체(사람, 위치), ② 장치와 선호도에 대한 개체(장치 정보, 장치 선호도), ③ 사회관계를 표현하는 개체(사람, 친구, 흥미).

지식의 재사용성은 온톨로지의 중요한 장점 중의 하나이다 (FOAF, 2006). 이를 위하여 CONON, FOAF, CC/PP 온톨로지를 추상 상황모델에 통합시켰다. <Figure 4>는 그중의 하나인 CC/PP 온톨로지를 보여주고 있다. 그리고 통합 및 확장된 모델은 <Figure 5>와 같다. 결국, 효과적으로 구축된 온톨로지 지식베

이스는 다른 온톨로지 지식베이스와 통합을 쉽게 지원하고, 새로운 지식을 창출할 수 있으며 이를 활용하여 많은 서비스들을 제공해 줄 수 있다.

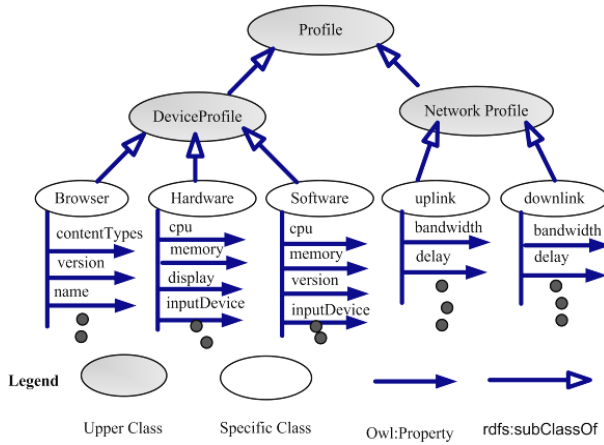


Figure 4. CC/PP ontology

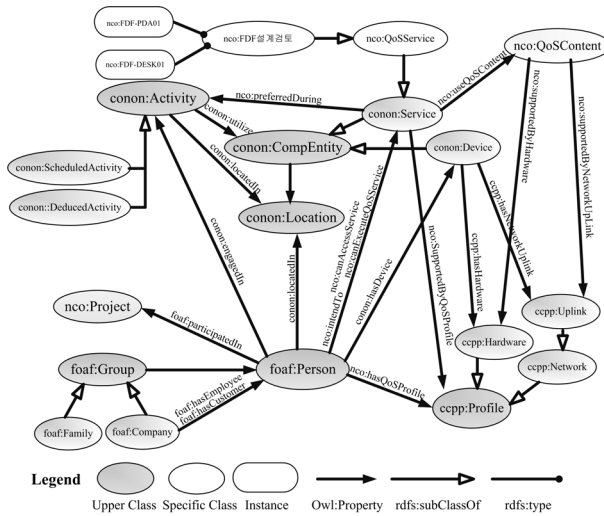


Figure 5. Integrated and extended ontology

3. QoS 제공을 위한 상황 추론

3.1 상황추론

상황은 매우 불확실하여 사용자와 관련된 상황을 토대로 사용자의 의도와 같은 불확실한 상황을 파악 하는 것은 매우 난해하다. 따라서 사용자 및 서비스 상황과 관련된 QoS 정보를 효과적으로 추론해 주는 것이 필요하다. 본 연구에서는 <Figure 6>과 같이 온톨로지 기반의 규칙베이스를 이용한 명시적인 상황추론 기법과 사용자의 의도와 같은 불확실한 상황정보를 추론하기 위한 신경망을 이용한 상황추론 방법을 함께 사용하여 보다 다양한 QoS를 제공한다. 특히, 온톨로지를 활용한 명시적인 상황처리를 위하여 Jena(2006)를 활용하였다.

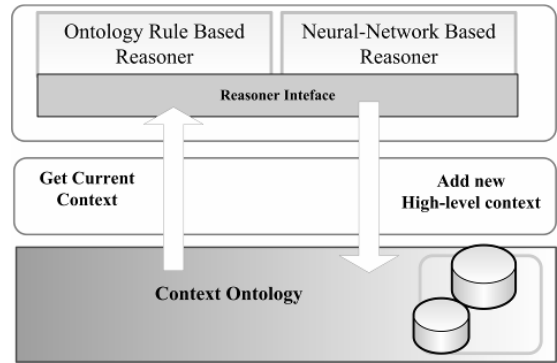


Figure 6. Context inferencing

아래는 온톨로지 기반의 상황추론 예를 보여준다. 여기서 사용자를 엔지니어로 근무하고 있는 용준이라고 가정하고, 특정 프로젝트에 참여하고 있는 용준이 휴일이 아닌 근무시간에 회사에 머물러 있다면 그는 현재 근무 중이라는 상황을 추론하게 된다. <Figure 7>~<Figure 9>는 용준의 현재 상황을 표현해주는 온톨로지와 그가 근무 중이라는 새로운 상황을 추론하는 규칙과 결과를 보여주고 있다.

```

<foaf : Person rdf : ID = "용준">
<foaf : name> 용준 </foaf : name>
<conon : locatedIn>
  <foaf : Company rdf : resource = "#IT기반 가상공학 서비스 회사"/>
</conon : locatedIn>
<conon : CurrentTime> 10a.m. </conon : CurrentTime>
<conon : Day> Monday </conon : Day>
<foaf : participateIn>
  <nco : Project rdf:resource = "#FDFproject"/>
</foaf : participateIn>
</foaf : Person>
    
```

Figure 7. Context ontology about "용준"

```

(notEqual(conon : Day,conon : Holiday))
(greaterThan(conon : CurrentTime, conon : StartWorkingTime))
(lessThan(conon : CurrentTime, conon : EndWorkingTime))
(?person conon : locatedIn foaf : Company)
(?person foaf : participateIn nco : Project)
-> (?person conon : engagedIn conon : Working)
    
```

Figure 8. An inferencing rule for a "Working" context

```

<foaf : Person rdf : id = "용준">
<conon : engagedIn>
  <conon : Activity rdf : resource = "#Working"/>
</conon : engagedIn>
</foaf>
    
```

Figure 9. Newly inferred "Working" context

온톨로지 규칙베이스의 활용은 명확한 상황에서는 효과적으로 사용될 수 있으나 추측하기가 어려운 상황정보들을 효과적으로 예측하여 활용하기는 어렵다. 따라서 온톨로지 지식베이스 기반의 신경망을 이용한 추론방법을 제안한다. 알고리즘 처리과정은 <Figure 10>과 같다.

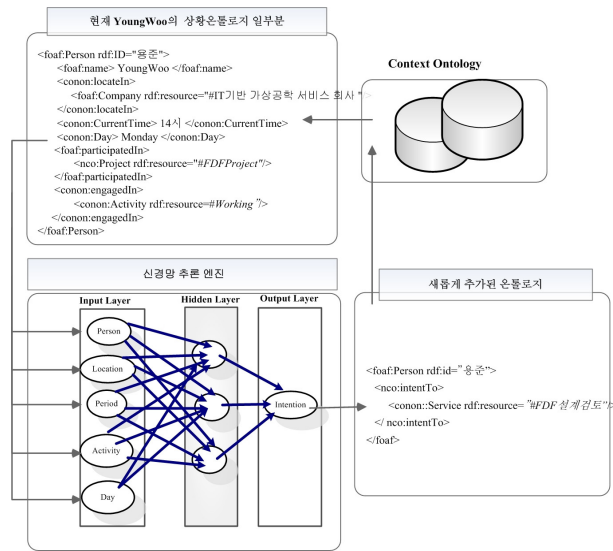


Figure 10. Inference procedure using a neural network

1. 온톨로지 지식베이스로부터 신경망을 구성하는 상황정보(사람, 위치, 시간대, 활동, 요일)를 얻어와 신경망을 구성한다.
2. 구성된 신경망을 이용하여 상황정보(의도)를 추론한다.
3. 상황정보(의도)를 추후에 상황인식 서버와 QoS 관리자가 이용할 수 있도록 온톨로지 지식베이스에 갱신해 준다.

예를 들어 용준이가 회사에서 14시에 근무 중 이라면 FDF 프로젝트에 대한 설계검토 서비스를 이용하고자 한다는 의도를 추론하는 것이다. 사용자의 의도를 추론하기 위해 온톨로지 지식베이스에서 결정력 있는 상황정보들을 얻어와 학습데이터로 이용한다. 여기서는 <Figure 10>과 같이 사람(Person), 위치(Location), 시간대(Period), 활동(Activity), 요일(Day)의 상황

Table 1. Experiment table

데이터 및 실험 조건	
학습데이터	60
테스트 데이터	10
신경망 가중치	0.5
학습률	0.6
모멘텀	0.9
데이터 및 실험 결과	
결과 인식률	87%

정보들을 온톨로지 지식베이스로부터 신경망의 입력 값으로 명시한다. 그리고 FDF 설계검토 서비스, 이메일 확인 서비스, 일정관리 서비스 등의 사용자가 원하는 의도를 반영한 서비스 정보들을 신경망의 출력으로 설정하여 학습시킨다. 본 논문에서 신경망 추론을 활용하기 위한 데이터 정보와 실험조건에 따른 결과는 <Table 1>과 같다.

<Table 2>와 <Table 3>은 신경망 학습을 위한 데이터 일부와 학습된 신경망을 통하여 명시된 입력 값에 따라서 추론된 일부 정보를 보여주고 있다.

Table 2. Data for training

학습데이터(입력 자료)					
사람	장소	시간대	활동	요일	의도
용준	Room1	10~11	Working	Monday	FDF 설계검토
동우	Room2	11~12	Working	Tuesday	FDF 설계검토
수영	Room3	9~9	Working	Thursday	Schedule02

Table 3. Inferred result

실제 데이터(입력 자료)					출력
사람	장소	시간대	활동	요일	의도
용준	Room2	13~14	Working	Friday	FDF 설계검토

이렇게 학습된 신경망을 통하여 새롭게 추론된 상황정보(의도)는 온톨로지 지식베이스에 갱신이 되고 차후에 사용자에게 QoS를 제공하기 위한 상황정보가 된다. <Figure 11>은 신경망이 능동적으로 추론한 온톨로지를 보여주고 있다. 이 온톨로지는 현재 상황에서 용준이가 FDF 설계검토에 대한 서비스를 이용하고 싶다는 의도를 추론한 것이다.

```
<foaf:Person rdf:id="용준">
  <nc:intendTo>
    <conon:Service rdf:resource="#FDF설계검토"/>
  </nc:intendTo>
</foaf>
```

Figure 11. Inferred context by applying a neural network

결과적으로, 대부분의 서비스는 상황 온톨로지 추론에 의해서 가능하지만, 부분적으로는 신경망 추론이 적용된다. 하지만, <Table 1>의 결과 인식률을 고려할 경우 상황에 따라 부정확한 추론을 야기시켜 사용자에게 잘못된 결과를 제공할 수도 있다.

3.2 QoS 상황추론

3.2.1 QoS Manager

유비쿼터스 환경하에서 사용자 중심의 서비스 제공을 실현

시키기 위해서는 장치자원(메모리, CPU 등)의 변화 또는 그와 관련된 네트워크 자원의(대역폭 등) 변화 그리고 선호도 등의 사용자와 관련된 QoS 상황정보를 최종적으로 고려해야 한다. 또한, 이를 기반으로 콘텐츠를 생성하여 사용자에게 제공해주어야 한다. 뿐만 아니라 유비쿼터스 환경은 변화가능성이 심하기 때문에 주기적으로 사용자와 관련된 QoS 상황변화를 감지하여 콘텐츠를 재구성해 줄 필요가 있다. 이를 위하여 본 논문에서는 QoS Manager를 제시한다. QoS Manager는 사용자의 서비스 QoS를 적절하게 고려하여 서비스를 제공하게 되는 데 이를 위한 각각의 컴포넌트와 각 역할은 <Figure 12>와 같다.

Communication Channel Module는 CAMUS의 PLANET(Kim et al., 2005)을 기반으로 재구성하였다. Communication Channel Module는 Context-Aware Server와 Agent들간의 또는 Agent와 Agent들간의 네트워크 연결을 제공하는 역할을 한다. 또한 이를 위하여 네트워크의 상태정보와 오류상황정보 들을 주기적으로 관리하게 되는데 이것은 변동성과 제약이 심한 네트워크 자원을 적절하게 이용할 수 있도록 해준다.

Content Generation Module는 사용자로부터 QoS 요청이 발생하면 기존의 사용자와 관련된 상황정보와 QoS 상황정보를 Context-Aware Server에 통보한다. Content Generation Module은 Context-Aware Server에 의해서 최종적으로 추론된 QoS 상황정보를 이용하여 그에 알맞은 콘텐츠를 선택하거나 생성하여 콘텐츠를 에이전트에게 제공해주는 역할을 한다. <Figure 12>는 이러한 구조 안에서 QoS Manager의 QoS 상황처리 절차를 보여주고 있다.

QoS 상황처리에 따른 콘텐츠 제공 절차를 살펴보면 Content Generation Module는 사용자로부터 QoS 요청이 발생 시 사용자 관련 상황정보와 이를 토대로 추론된 사용자의 의도와 사용자의 활동 등의 상황정보를 Context-Aware Server로부터 얻어온다.

여기서 Content Generation Module은 사용자와 관련된 QoS 상황 정보 온톨로지를 구성해주기 위하여 Context-Aware Server로부터 얻어온 상황정보와 QoS 상황정보를 합성하여 QoS 상황정보 온톨로지를 구성한다. 다음으로 Content Generation Module은 구성된 QoS 상황정보 온톨로지를 Context-Aware Server에 보내어 QoS 상황정보를 재구성하도록 한다. QoS 상황정보를 재구성하는 과정에서 Context-Aware Server는 QoS 상황정보들을 추론하여 상황 온톨로지를 구성한다.

마지막으로 Content Generation Module은 QoS를 고려한 서비스 추론 결과를 이용하여 적합한 콘텐츠를 선택하거나, 조율하여 Service Agent에게 제공해 준다. 또한 서비스 이용도중 서비스 이용과 관련된 사용자의 상황이 바뀌었을 경우 현재 사용하는 콘텐츠의 품질속성을 다시 재구성해주는 역할을 하게 된다. 예를 들면, 사용자가 서비스를 이용하고 있는 도중 다른 서비스의 실행으로 인하여 장치의 메모리가 부족하게 될 경우나 특정 서비스의 선호도 변화와 같은 품질과 관련된 상황변화가 발생시 상황에 적합하도록 콘텐츠를 재구성하여 제공한다.

3.2.2 QoS 제공 예

아래에 제시된 시나리오를 통하여 어떻게 QoS 상황정보를 추론하고 QoS를 제공해줄 수 있는지 보여주고자 한다. 현장에 나와 있는 동우 직원은 회사에 FDF 설계업무를 담당하고 있는 용준 직원과 FDF 프로젝트에 대한 설계 자료를 검토하기를 원한다. 여기서 동우 직원은 저사양의 PDA 장치만 이용할 수 있어서 고품질의 FDF 설계검토 서비스를 이용할 수 없기 때문에 이에 맞도록 서비스 품질을 조절하여 FDF 설계검토 서비스를 제공받아야 한다. <Figure 13>은 이를 위하여 QoS를 고려한 서비스를 제안해주고 있는 규칙과 인스턴스에 따른 결과를 보여 주고 있다. 이는 현재 동우 직원이 이용할 수 있는 PDA의 메모리 크기,

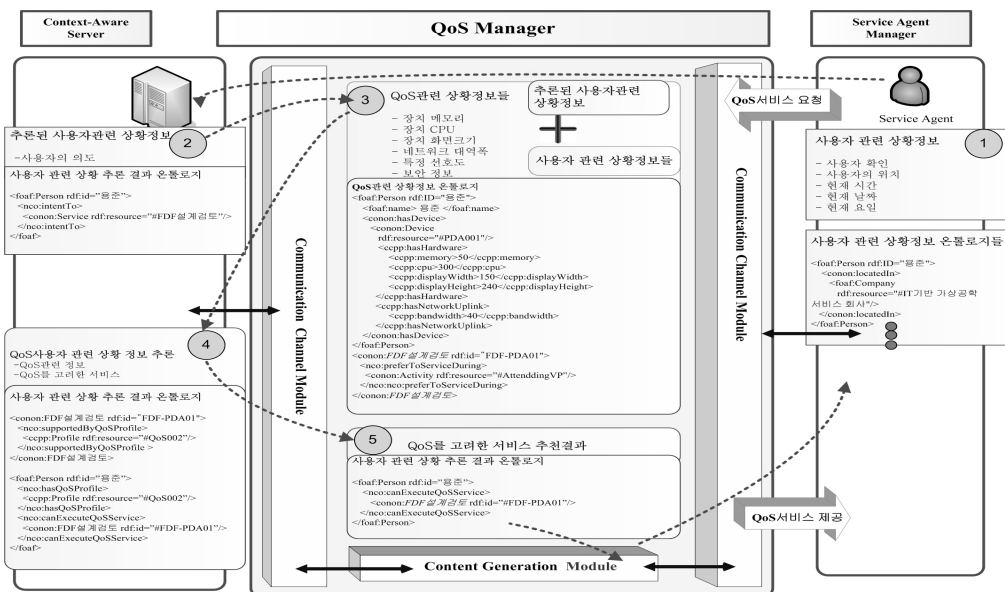


Figure 12. QoS context processing procedure in the QoS manager

CPU 성능, 디스플레이 장치의 크기, 네트워크의 상황정보를 토대로 하여 품질에 알맞은 FDF 설계검토 서비스를 추천해 준다.

```
(?person conon : hasDevice ?device)
(?device ccpp : hasHardware ?hardware)
(?hardware ccpp : memory ?memory)
(?hardware ccpp : cpu ?cpu)
(?hardware ccpp : displayHeight ?displayHeight)
(?hardware ccpp : displayWidth ?displayWidth)
(?device ccpp : hasNetworkUplink ?link)
(?link ccpp : bandwidth ?bandwidth)
( lessThan(?memory, 100))
( lessThan(?cpu, 330))
( lessThan(?displayHeight, 240))
( lessThan(?displayWidth, 150))
( lessThan(?bandwidth, 40))
->( ?person nco : hasQoSProfile ccpp : QoS002)

(?service nco : useQoSContent ?qoscontent)
(?qoscontent nco : supportedByHardware ?hardware)
(?hardware ccpp : memory ?memory)
(?hardware ccpp : cpu ?cpu)
(?qoscontent nco : supportedByNetworkUplink ?link)
(?link ccpp : bandwidth ?bandwidth)
( lessThan(?cpu, 330))
( lessThan(?memory, 100))
( lessThan(?displayHeight, 240))
( lessThan(?displayWidth, 150))
( lessThan(?bandwidth, 40))
->( ?service nco : supportedByQoSProfile ccpp : QoS002)

(?person nco : hasQoSProfile ?profile)
(?person nco : intendTo ?service)
(?qos-supported-service rdfs : type ?service)
(?qos-supported-service nco : supportedByQoSProfile ?profile)
->( ?person nco : canExecuteQoSService ?qos-supported-service)
```

```
<foaf : Person rdf : id = "용준">
  <nco : hasQoSProfile>
    <nco : Profile rdf : resource = "#QoS002"/>
  </nco : hasQoSProfile>
  <nco : intendTo>
    <conon : Service rdf : id = "FDF 설계검토"/>
  </nco : intendTo>
  <conon : engagedIn>
    <conon : Activity rdf : resource = "#Working"/>
  </conon : engagedIn>
</foaf>
<nco : FDF 설계검토 rdf : id = "FDF-PDA01">
  <nco : supportedByQoSProfile>
    <nco : Profile rdf : resource = "#QoS002">
      </nco : supportedByQoSProfile >
  </nco : FDF 설계검토 >
<foaf : Person rdf : id = "용준">
  <nco : canExecuteQoSService>
    <nco : FDF설계검토 rdf : id = "FDF-PDA01"/>
  </nco : canExecuteQoSService>
</foaf>
```

Figure 13. QoS-based service recommendation applicable to PDA

규칙의 표현 방식에 따라서 서비스의 QoS 정보를 선택하는 방법은 다양하게 존재할 수 있다. MPEG, JPEG 등의 압축방식의 영향을 많이 받는 멀티미디어 서비스의 경우 QoS에 따라서 멀티미디어의 FrameRate, Resolution 등의 멀티미디어 정보와 같은 내부적인 속성에 따라서 서비스를 조절하여 제공해 줄 수 있다. 경우에 따라서는 다양하게 제공되고 있는 몇 가지 QoS 속성을 가지고 있는 콘텐츠들 중에서 QoS에 적합한 콘텐츠를 선택하여 제공해 줄 수도 있을 것이다.

정보유출이 심한 환경에서는 서비스의 보안을 제공해야 한다. <Figure 14>는 보안을 고려한 서비스 제공 방법을 보여주고 있다. 제시된 보안규칙은 모든 사람들에게 서비스가 이루어지지 않고 제한된 사용자만 서비스를 제공받을 수 있다는 것을 정의하고 있다. 이러한 보안규칙은 설계자료를 수정할 수 있는 FDF-DESIGN01을 CiSCompany 회사내부에서 근무하는 FDF Engineer로 국한하고 있다.

```
(foaf : CiSCompany foaf : hasEmployee ?person)
  (?person foaf : hasPosition foaf : FDFEngineer)
  (?person conon : locatedIn foaf : CiSCompany)
  ->( ?person nco : canExecuteQoSService conon : FDFDESIGN01)
```

Figure 14. Security rule

4. 시스템 구현과 실험평가

4.1 구현

본 장에서는 유비쿼터스 및 상황인식 컴퓨팅 환경하에서 제시된 프레임워크가 어떻게 사용자 중심의 QoS를 제공할 수 있는지에 관해서 구체적으로 설명한다. 특히, PDA에서 서비스를 지원하기 위해서 PDA용 Java 가상머신인 Crème를 사용하였으며 ARToolkit(2006)을 증강현실 구현도말로 활용하여 시나리오를 구현하였다. 제시된 방법은 <Figure 15>에서 볼 수 있듯이

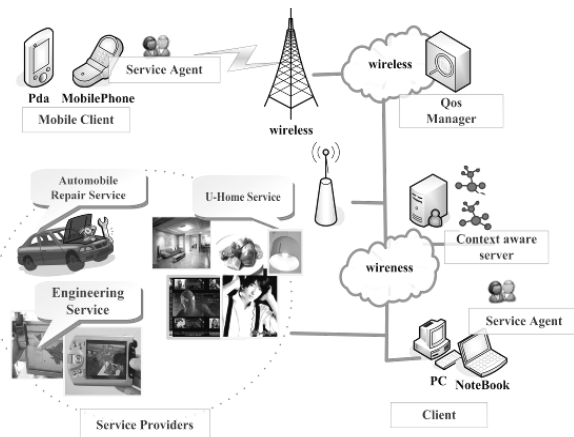


Figure 15. QoS-based service scenarios

다양한 분야에 적용이 가능하며, 특히, 엔지니어링 협업 서비스, 자동차 정비 서비스(Rhee *et al.*, 2006), 유비쿼터스 홈 서비스 등을 제공하는 응용시스템을 구현하여 제시된 방법의 실현가능성을 테스트하였다.

<Table 4>과 <Table 5>는 이러한 서비스를 제공하는 과정에서 디바이스의 QoS 상황을 어떻게 처리하는지를 구체적으로 설명하고 있다. <Table 4>는 용준의 현재 상황과 이용 가능한 서비스들의 온톨로지 구조를 보여주고 있다. <Table 5>는 추천된 QoS에 대하여 구체적으로 설명하고 있다. <Figure 16>은 추천된 QoS 구현결과를 보여주고 있다.

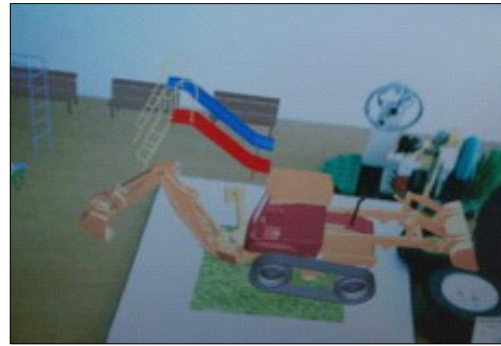
Table 4. Context description of “용준” and ontology structure of the available services

용준의 현재 상황
현장에 나와 있는 용준은 사무실에 있는 동우와 FDF 프로젝트의 설계에 관하여 협업하고자 한다. 하지만 현장에 있는 용준은 무선 네트워크를 이용하는 저사양의 PDA 장치만 이용할 수 밖에 없다. 하지만, 이들간의 효과적인 협업이 이루어지기를 시스템이 지원해 주기를 원한다.
이용 가능한 서비스들의 Ontology 구조



Table 5. Recommended service description

추천된 FDF-PDA01서비스에 대한 설명	
하드웨어 권장사항	CPU : 300MH , 메모리 : 100MB
네트워크 권장사항	네트워크 대역폭 40MBPS
QoSContent	현재의 PDA 사양으로는 직접 3D 가시화를 구현할 수 없기 때문에 서버에서 3D로 구현한 후 결과를 사용자의 디바이스 상태에 적합한 형태로 전송하는 방식으로 서비스를 제공한다. 예를 들면, 3D 모델을 JPEG 기반의 이미지와 네트워크 효율성을 높이기 위하여 UDP 기반의 RTP 프로토콜을 이용하여 정보들을 전송한다.
컨텐츠 내용	네트워크상에 있는 사용자와 FDF 설계에 관해서 검토하고 논의할 수 있는 3D기반의 설계자료와 패널형식의 회의 서비스를 제공하는 엔지니어링 협업 서비스이다.



(a)



(b)

Figure 16. Collaborative engineering service

<Figure 17>은 QoS를 고려한 자동차정비 서비스를 제공하는 화면을 보여주고 있다. <Figure 18>는 QoS를 고려한 유비쿼터스 홈 서비스를 제공하는 화면을 보여주고 있다.

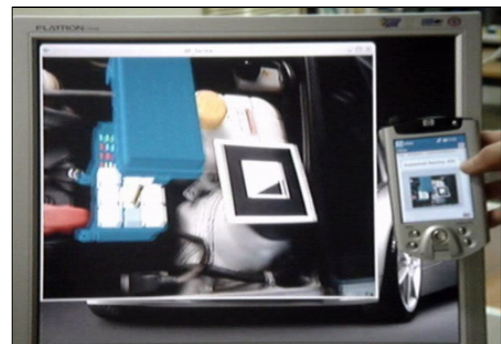


Figure 17. Car maintenance service

4.2 실험 평가

본 논문에서 제안된 상황인식 프레임워크의 유용성 평가를 위해 엔지니어링 협업서비스, 자동차 정비 서비스, 유비쿼터스 홈서비스를 구현하고 이를 모의 환경에 각각 설치하였다. 환경 내부에는 사용자의 상황을 수집하고 서비스를 제공하는 PC기반의 에이전트, 상황 관리와 추론 등을 담당하는 상황인식 서버가 제공되었다. 마지막으로 서비스 실행을 위한 에이전트가 PC와 PDA에 설치되었다. 제공된 상황인식 프레임워크



Figure 18. Ubiquitous home service

의 유용성 평가를 위해서 15명을 대상으로 실험을 수행하였다. 구현된 시스템 사양은 Intel Xeon 2.6 GHz(Dual CPUs), 1Gbytes (Memory)와 Window XP(OS)가 사용되었다. PDA는 Intel Xscale PXA 250, 240×320(Display), 100MByte(Memory), 16Bit(Color)가 사용되었다.

<Table 6>에서 볼 수 있듯이 실험내용은 상황인식 기반의 서비스 제공을 위한 QoS 상황을 고려하지 않은 기존의 방법과 QoS 상황을 고려한 제안된 서비스에 대해서 비교하였다. 실험은 참여자의 실험결과와 설문조사를 바탕으로 상호작용 효율성, 서비스 만족도, 지연시간, 서비스 리소스 점유율을 비교 및 측정하였다. 상호작용 효율성과 서비스 만족도는 100점 만점을 기준으로 피실험자의 평가에 근거를 두었으며 지연시간과 서비스 리소스 점유율 등의 정보는 100점 만점을 기준으로 실제 시스템의 실행정보를 참조하여 점수를 계산하였다. 결과에 따르면, 특히, 상호작용 효율성과 서비스 만족도에서 많은 차이가 발생하였다. 즉, QoS 상황을 고려한 서비스가 보다 나은 상호작용 효율성을 제공함으로써 사용자에게 보다 큰 서비스 만족도를 보여주었다.

Table 6. Experiment Result

	OoS 활용여부에 따른 실험평가 결과					
	엔지니어링 협업 서비스		자동차 정비 서비스		유비쿼터스 홈 서비스	
	A	B	A	B	A	B
상호작용 효율성	49	67	50	75	53	75
서비스 만족도	67	80	61	84	59	82
지연시간	34	33	37	31	35	30
서비스 리소스 점유율	40	42	40	41	40	43

주) A : QoS상황을 고려하지 않은 서비스.

B : QoS 상황을 고려한 서비스.

QoS 상황을 고려한 서비스는 디바이스 성능, 사용자 선호도, 보안 등의 QoS 상황에 따라서 원활한 서비스를 제공할 수 있었다. 또한 네트워크 기반의 상호작용 시에는 사용자의 무선 환

경의 대역폭에 따라 대역폭에 맞는 통신 프로토콜, 전송시간을 설정하여 서비스를 이용할 수 있어서 상호작용 효율성을 높여주었다. 또한, 특정 상황에서 사용자의 선호도에 따라서 차별화된 서비스를 제공할 수 있었다.

QoS 상황을 고려하지 않는 서비스는 모든 사용자에게 품질은 고려하지 않은 동일한 서비스를 제공하여 특정 장치에서는 돌아가기 힘든 서비스를 제공해주는 경우가 발생하였다. 또한 네트워크기반의 상호작용 시에는 네트워크 대역폭 등을 고려하지 않는 프로토콜, 코덱 등을 이용함으로써 실행 과부하가 발생하여 서비스의 질을 저하시켜 상호작용 효율성을 경감시켰다. 뿐만 아니라, 장치가 지원하지 못하는 해상도, 기타 입/출력 장치에서 실행 가능한 서비스가 제공되어 사용자의 불편함을 초래한다고 평가하였다. 결국, 다양한 QoS 상황 정보를 활용하여 개인마다 차별화된 서비스를 제공할 수 있어서 사용자는 자신의 취향과 의도와 운용하고 있는 장치에 맞는 맞춤형 서비스를 받을 수 있어 만족스럽다는 평가를 받았다.

5. 결론

본 논문에서는 유비쿼터스 환경하에서 다양한 서비스들을 사용자의 환경에 적합하게 제공하기 위한 방안으로 상황인식기반의 QoS 제공방법을 제안하였다. 또한, 구체적인 구현 결과를 제시하였다.

온톨로지를 이용하여 상황지식 간의 상호 운용성을 높였으며, 특히, 사용자의 선호도, 보안 등을 지원할 수 있도록 상위 온톨로지를 확장하여 다양한 환경에서 다양한 형태의 맞춤형 서비스를 지원할 수 있도록 처리하였다. 또한, 온톨로지 규칙과 신경망을 이용하여 사용자와 관련된 상황과 서비스에 관련된 QoS 상황정보들을 관리 및 추론하여 사용자의 상황에 적합한 QoS를 제공할 수 있었다.

하지만 본 논문에서 제안한 상황정보 추론방법이 부정확한 경우 부적절한 서비스가 제공되는 경우가 발생하게 된다. 또한, 제공된 시스템이 프로토타입 수준으로 개발되어 활용성에 대한 다양한 가능성을 보여주기 위한 더 많은 연구 및 실험평

가가 필요하다. 특히, 신경망의 인식률을 더 높일 수 있는 추가적인 연구가 필요하다. 뿐만 아니라, 정확한 추론방법과 예외적인 상황처리를 위한 방법을 연구해야 하며, 다양한 상황에서 발생할 수 있는 사용자들 간의 충돌, 자원들 간의 충돌 등의 상황충돌 문제를 연구해야 한다.

참고문헌

- Alber, H., Sven, B., and Alexander, S. (2002), Modeling of context information for pervasive computing applications, www.rn.inf.tudresden.de/scripts_lsrn/veroeffent_print/SCI2002-paper512JH.pdf.
- ARToolkit (2006), www.hitl.washington.edu/artoolkit/.
- Bisdikian, C., Boamah, P., and Catro, A. (2002), Intelligent pervasive middleware for context-based and localized telematics services, *Proc. of the 2nd International Workshop on Mobile Commerce*, 15-24.
- CC/PP (2004), www.w3.org/TR/CCPP-struct-vocab/.
- Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F., and Chakraborty, D. (2004), Intelligent agents meet the semantic web in smart spaces, *IEEE Internet Computing*, 8(6), 69-79.
- CrEme, <http://www.nsicom.com/>.
- Dey, A. K. and Abowd, G. D. (2000), The context toolkit : aiding the development of context-aware applications, *Workshop on Software Engineering for Wearable and Pervasive Computing*.
- Fano, A. and Gershman, A. (2002), The future of business services in the age of ubiquitous computing, *Communications of the ACM*, 45(12), 83-87.
- FOAF (2006), <http://xlns.com/foaf/0.1/>.
- Henricksen, K., Indulska, J., and Rakotonirainy, R. (2002), Modeling Context Information in Pervasive Computing Systems, *LNCS (Pervasive 2002)*, 2414, 79-117.
- Jena (2006), <http://jena.sourceforge.net>
- Kim, H., Cho, Y. J., Oh, S. R. (2005), CAMUS : A middleware supporting context-aware services for network-based robots, *IEEE Advanced Robotics and its Social Impacts*, 237- 242.
- Rhee, G. W., Seo, D. W., and Lee, J. Y. (2006), Maintenance simulation service using augmented reality in ubiquitous environments, *Proc. of 11th Society of CAD/CAM Engineers Conf.*, 302-308.
- Wang, X., Dong, J. S., and Chin, C. Y. (2004), Semantic space: an infrastructure for smart spaces, *IEEE Pervasive Computing*, 3(3), 32-39.
- Weiser, M. (1991), The computer of the 21st century, *Scientific American*, 66-75.
- Yau, S. S., Karim, F., Wang, Y., Wang, B., and Gupta, S. (2002), Reconfigurable context-Sensitive middleware for pervasive computing, *IEEE Pervasive Computing*, 1(3), 33-40.



서동우

전남대학교 산업공학과 학사
 전남대학교 산업공학과 석사
 현재: 전남대학교 산업공학과 박사과정
 관심분야: Context-Aware Computing,
 Information System Modeling,
 Software Architecture



이재열

포항공과대학교 산업공학과 학사
 포항공과대학교 산업공학과 석사
 포항공과대학교 산업공학과 박사
 한국전자통신연구원 선임연구원
 현재: 전남대학교 산업공학과 부교수
 관심분야: Tangible Interface, Augmented and
 Virtual Reality, Context-aware
 Computing, Ubiquitous Computing,
 Collaborative Virtual Engineering