유비쿼터스 컴퓨팅 환경에서 Event Driven 상황정보 모델링 및 서비스 구현

Context aware Modeling and Services Implementation With Event Driven in Ubiquitous Computing Environments

김 형 선* 김 현** 문 애 경*** 조 준 면**** 홍 충 성*****
HyoungSun Kim Hyun Kim Aekyung Moon JunMyun Cho ChungSung Hong

요 약

최근 상황인식 기술은 유비쿼터스 환경에서 중요한 이슈로 떠 오르고 있으며, 연구자들에 의해 많은 연구가 이루어 지고 있다. 상황인식 기술은 상황정보를 획득/가공하여 사용자에게 필요한 서비스와 정보를 제공하는 컴퓨터 시스템이다. 상황인식 시스템은 각종 센서로부터 발생하는 낮은 수준의 상황정보를 높은 수준의 상황정보로 변환하여 사용자에게 원하는 서비스를 제공하는 환경을 만들어 준다. 본 논문에서는 물리적인 스마트 공간에 존재하는 각종 센서로부터 얻어진 하위단계(Low level)의 Event driven 센서정보를 온톨로지 기반 상황정보로 모델링 하여 상위단계(High level) 상황정보로 변환하는 상황인식 모델링과 시스템을 제안한다. 또한 변환된 High level 상황정보는 가상공간에서 ECA(Event, Condition, Action) 룰에 따라 사용자에게 언제 어디선지 사용자가 필요로 하는 능동적이고 지능적인 서비스를 제공하고, 스마트 오피스 환경에서 발표 도우미 서비스를 구현하였다.

Abstract

Context aware computing is an emerging paradigm to achieve ubiquitous computing environments by enabling computer systems to understand their situational contexts. A context aware system uses context to provide relevant information and services to the user depending on the user's task. In this paper, we propose an ontology based context aware modeling methodology that transmits low level contexts acquired by directly accessing various sensors in the physical environments to high level contexts. With these high level contexts, context aware application can provides proactive and intelligent services using ECA (Event Condition Action) rules. We implemented a presentation service in smart office environment.

☞ Keyword : Context-aware, Ubiquitous Computing, event-driven, ontology

1. 서 론

유비쿼터스 환경에서 능동형 서비스를 제공하기 위한 기본적인 요구조건은 어떠한 이질적인 환경에서도 사용자에게 유연하게 서비스를 제공할 수 있어야 하고, 현재 서비스 환경 내에 존재하는 모든 센서와 장치들간에 상호운용성을 요구한다.

이러한 요구를 충족하기 위한 물리적인 스마트 공간은 내장된 센서, 가전기기, 컴퓨터, 모바일 장치 같은 다양한 정보자원이 위치하고 있는 환경에서 상황정보를 제공하기 위하여 사용되는 공간이다.

상황인식 어플리케이션은 물리적 스마트 공간에서 동작하고 있는 센서들로부터 상황정보를 획득하여 상황인식을 할 수 있다. 물리적인 스마트 공간에서 각종 센서들을 자동으로 찾아서 상황정

* 정 회 원 : 한국전자통신연구원 소프트웨어로봇연구팀 책임연구원 kimhs@etri.re.kr
** 정 회 원 : 한국전자통신연구원 소프트웨어로봇연구팀장, 책임연구원 hyunkim@etri.re.kr
*** 정 회 원 : 한국전자통신연구원 소프트웨어로봇연구팀 선임연구원 akmoon@etri.re.kr
**** 정 회 원 : 한국전자통신연구원 소프트웨어로봇연구팀 선임연구원 jmcho@etri.re.kr
***** 정 회 원 : 한국전자통신연구원 소프트웨어로봇연구팀 연구원 cshong@etri.re.kr
[2006/04/24 투고 - 2006/05/30 심사 - 2006/08/30 심사완료]

보를 획득할 수 있고, 사용자에서 능동적인 서비스를 제공하기 위해 지식능력을 가진 상황정보로 변경할 수 있다. 그러나 상황인식 어플리케이션은 센싱, 표현, 합성, 저장, 질의, 상황을 추론하는 것과 관련된 광범위한 분야를 다루기 때문에 개발자들이 구현하는 것이 쉽지 않다.

이런 문제를 해결하기 위하여 물리적 스마트 공간에서 상황정보의 명백한 표현, 의미 있는 질의, 상황정보의 유연한 추론을 위하여 시맨틱 웹 기술을 이용하여 온톨로지 기반 상황인식 구조인 가상 공간을 제안하고, 필요한 서비스를 구현하기 위하여 상황인식 어플리케이션을 구현 하였다. 가상공간에 대한 구조를 정의하고 난 뒤, 재사용이 가능한 컴포넌트로 일반적인 상황인식 메커니즘을 제공함에 따라 어플리케이션 개발자들은 가상공간에서 상황인식 어플리케이션을 구현하는데 있어서 비용을 절감하고 복잡성을 줄일 수 있도록 하였다.

상황인식 하부구조로써 제공된 가상공간의 중요한 기능은 각종 센서로부터 수집된 낮은 수준의 정보를 높은 수준의 상황정보로 유도해 내는 기능이다. 상황인식 시스템에서는 사용자의 상태와 주변환경에 대한 요약된 높은 수준의 상황정보를 이용하여 사용자에게 상황에 맞는 서비스를 제공할 수 있다[1].

높은 수준의 상황정보는 각종 하드웨어 센서와 소프트웨어 프로그램에서 직접적으로 획득한 낮은 수준의 정보로부터 유추한 것이다.

예를 들면 침실에서 사람이 자고 있다는 높은 수준의 상황은 침실에 장착된 각종 센서에 의해서 제공되는 낮은 수준의 상황정보로부터 유추해 낸 것을 알 수 있다.

즉, (1)사람은 침실에 있다. (2)침실의 조명은 어둡다. (3)침실의 소음도는 낮다. (4)침실 문은 닫혀져 있다는 낮은 수준의 상황정보부터 "사람이 침대에 자고 있다"는 높은 수준의 상황정보를 유추해 낸 것이다.

상황인식 하부구조에서의 상황정보는 온톨로지 기반의 상황정보 모델로 모델링 되었고, 상황정보 지식 베이스로 유지 관리한다. 높은 수준의 상황정보는 상황정보를 저장한 지식 베이스에서 규칙 기반 추론기술을 적용함으로써 유도할 수 있다. 상황정보를 추론하고, 검증, 질의하는 상황인식 하부구조의 장점은 상황정보가 기계가 해석할 수 있는 형태로 명확하게 표현 되었기 때문에 일반적인 메커니즘으로 쉽게 지원할 수 있다.

본 논문에서 제안하는 상황인식 시스템은 event driven 분산 모델을 근간으로 하는 가상공간에서 상황인식을 하기 위한 방법을 제안하고, 상황표현, 추론, 검증, 질의를 위하여 가상공간상에서 획득한 상황정보를 가공하여 사용자에게 상황에 맞는 능동형 서비스를 제공하기 위한 서비스를 구현한다[2].

Event driven 기반 상황정보 추론은 상황정보 모델링을 통하여 구현되고, 가상공간상에서 동적으로 상황정보를 획득하고 표현하는 방법을 제공할 뿐만 아니라 가상공간에서 사용하기 위한 어플리케이션에 다양하고 광범위한 높은 수준의 상황정보를 제공하기 위하여 상황인식 하부구조에 상황정보 추론 메커니즘을 도입하여 쉽게 통합할 수 있다.

따라서 이러한 요구를 충족하기 위하여 온톨로지 기반 상황정보를 모델링한 것을 이용하여 스마트 오피스 환경에서 발표 도우미 서비스를 구현하고 간단한 실험을 하였다.

본 논문의 구성은 다음과 같다. 2장의 관련 연구에서는 기존의 상황인식 기반 응용 서비스에 대하여 언급하고 3장에서는 정의한 가상공간에 대하여 상세하게 설명한다. 4장에서는 본 논문에서 제안하는 하위단계의 상황정보를 상위단계의 상황정보로 변환 하기 위한 온톨로지 상황정보 모델링에 대하여 설명하고, 5장에서는 구체적인 실험 및 구현에 대하여 설명하고 마지막으로 6장에서 결론을 맺는다.

2. 관련연구

상황인식 컴퓨팅은 약 10년 전에 제안된 이래 많은 학자들이 연구하여 왔으며, 기술의 유용성을 입증하기 위해 다수의 상황인식 응용 프로그램이 구축 되었다. 다음은 이미 개발된 몇몇의 상황인식 응용에 관한 대략의 설명이다.

- Call Forwarding[3] : Olivetti Research Ltd. 에서 개발한 것으로 센서정보로 알아낸 사용자 위치를 상황정보로 사용한다. 이것은 사용자의 위치를 인식해서 전화가 왔을 때 사용자로부터 가장 가까운 곳에 있는 전화기로 연결하여 준다.
- CyberGuide[4] : Georgia 기술 연구소에서 개발한 것으로 관광객의 위치와 시간을 상황정보로 이용한다. 다양한 종류의 시스템이 사용자에게 현재 위치에서 이용 가능한 정보 서비스를 제공한다. 또한 사용자가 이전에 여행한 이력을 자동으로 편집하여 새로운 여행지를 추천하는데 이용한다.
- Conference Assistant[9] : Georgia 기술 연구소에서 개발한 것으로 이 시스템은 회의 참석자를 돕기 위해 다양한 상황정보를 이용한다. 회의일정, 사용자 위치, 사용자 관심분야 등을 조사하고 사용자가 발표장에 들어가면 자동으로 발표자의 이름과 발표주제 및 다른 관련정보를 보여준다.
- People and Object Pager[8] : Kent 대학에서 개발한 것으로 상황정보 사용자 위치, 근처의 사람과 사물을 이용한다. Page는 방문객으로부터 가장 가까이에 위치한 사람에게 메시지를 라우팅 함으로써 페이징 장치를 갖고 있지는 않지만 액티브 배지를 가진 방문객에게 메시지를 보낼 수 있다.

위의 연구에서 이용한 상황정보는 전형적인 사용자 입력형태와는 달리 여러 종류의 다양한 센서로부터 정보가 입력된다. 따라서 입력정보인 상황정보에 대한 추상화가 요구된다. 또한 물리적인 공간에서 상황정보를 센싱하는 컴포넌트와 가상공간에서 사용자에게 서비스를 제공하는 응용시스템과 결부되어 있지 않고 완전히 분리되어 있으며, 상황정보를 처리하기 위한 서비스와 장치를 동적으로 찾을 수 있어야 한다.

상황인식 모델링에 관련된 연구는 정형적인 모델링과 비정형적인 모델링 방법이 제안 되었다. 비정형적인 모델링 방법은 스키마[14,15]를 근간으로 모든 것을 표현 하는데, 비정형적인 모델링으로 된 시스템인 상황정보 툴킷[1]은 속성 값으로 표현하는 튜플(tuple)의 형태 상황지식을 표현하였고, Henrickse et al.[17]은 엔티티와 엔티티와의 관계와 UML을 사용하여 상황정보를 표현하였다.

정형적인 모델링 방법으로는 상황정보를 추론하는 단계까지 기술하기 위하여 정형적인 상황정보를 표현방법으로 접근하였다. Ranganathan et al.[18] 은 Gaia 시스템에서 DAML + OIL에서 first order 로직을 사용하여 상황정보를 표현하였고, Wang et al.[20] and Chen et al.[19]은 OWL를 사용하여 상황정보를 표현 하였다.

3. 가상공간의 정의

실 세계(Real World)는 물리공간 (Physical Space)과 가상공간(Virtual Space)으로 구성된다. 물리공간은 실제적인 물리객체(Physical Object)들로 구성되어 있으며[9], 사람은 이러한 물리객체와의 상호작용을 통해 일을 한다. 가상공간은 물리공간을 추상화 것으로써, 물리객체들을 전산 자원(Electronic Resource)들로 매핑(Mapping)하여 모델링 된다. 물리공간과 가상공간을 매핑하는 수단은 구현된 랩퍼를 통해 상호작용이 가능하다[12].

예를 들어, 물리공간에서 네트워크 통신이 가능한 센서와 제어기가 있다면, 이 센서와 제어기는 가상공간에서 센서 객체와 제어기 객체로 존재할 수 있다. 물리공간에서의 센서가 획득한 정

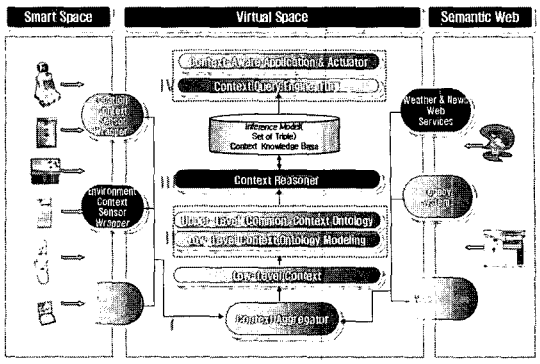
보는 가상공간에서의 센서 객체로 전달될 수 있으며, 또한 가상공간에서의 제어기 객체의 속성 중 전원값을 ON으로 변경함으로써, 물리공간의 제어기에 전원을 ON시킬 수 있게 된다. 만약 물리공간에서 센서와 제어기가 부착되어 있다면, 또 하나의 물리 공간으로 존재하며, 센서와 제어기는 가상 공간 상의 센서 객체와 제어기 객체로 존재하게 된다.

본 논문에서는 실제 물리 공간을 추상화한 것을 가상 공간이라고 정의하며, 공간영역 (Place)의 포함 관계에 따라 계층적인 구조를 갖게 되며, 각 영역 내에는 다양한 센서나 기기 등의 전산 자원을 갖는다. 실제 물리공간 내 존재하는 각종 센서나 기기 등의 물리자원은 가상공간에서 그 기능을 수행할 수 있도록 전산자원으로 만들기 위하여 물리적인 공간에 존재하는 자원을 가상공간에서 사용하기 위하여 똑같이 모델링 한다[2].

(그림1)은 온톨로지 기반 상황인식 하부구조인 가상 공간의 구조를 보여주고 있으며, 가상공간은 물리공간인 스마트 공간의 여러가지 다양한 센서로부터 상황정보를 수집하기 위하여 위치, 환경, 디바이스 컴포넌트가 서로 협업하는 상황정보 센서 랩퍼(Contex Sensor Wrapper), 상황정보 수집기(Context Aggregator), 상황정보 지식베이스(Context Knowledge Base), 상황정보 추론기(Context Reasoner)와 상황정보 질의엔진(Context Query Engine)으로 구성되어 있다.

(그림 1)에서 보는 바와 같이 물리적인 스마트 공간에서 상황정보 랩퍼는 하드웨어 센서나 소프트웨어 프로그램 같은 다양한 자원으로부터 하위 수준의 상황 정보를 획득해서 마크업 언어 변환한다. RFID와 같은 위치 상황정보 랩퍼(Location Context Wrapper)와 내장된 센서로부터 온도, 소음, 습도 같은 환경적인 정보를 획득하는 환경 상황정보 랩퍼(Environment Context Wrapper)는 물리적인 환경에 배치된 하드웨어 센서가 작동한다.

시맨틱 웹 공간에서 활동하는 상황정보 랩퍼(Activity Context Wrapper)는 마이크소프트 아웃룩



(그림 1) 가상공간에서의 상황인식 프레임워크

2000으로부터 일정정보를 추출할 수 있고, 날씨 상황정보 랩퍼(Weather Context Wrapper)는 지역 날씨정보를 얻기 위하여 날씨 웹 서비스를 호출한다.

또한 상황정보 센서 랩퍼(Context Sensor Wrapper)는 물리공간의 센서를 가상공간으로 맵핑하고, 실세계 환경의 변화 및 상황은 센서들에 의해 감지되어 상황정보 수집기(Context Aggregator) 계층을 통해 상위 계층에 전달된다.

상황정보 수집기(Context Aggregator)는 분산된 랩퍼를 발견하여 감지된 환경 데이터를 수집하여 온톨로지 기반 상황정보 모델로 모델링 되어 상황정보 지식 베이스에 저장 한다. 상황정보 수집기 계층은 물리 공간의 센서를 가상 공간으로 매핑하고, 이들 센서 정보로부터 상황 데이터를 추출하여 온톨로지 기반 상황정보 모델로 변환함으로써 상황인식 어플리케이션이 능동적으로 서비스를 제공할 수 있도록 지원한다[9,10]. 사용자로부터 음성정보, 영상정보, 온도/습도 정보, 사용자 일정정보 등이 모두 센서 정보가 될 수 있다. 이 계층에서는 상황 정보의 추출뿐만 아니라 이들을 가공(interpretation)하는 역할도 수행한다. 예를 들어, 센서 정보를 필터링 하거나 조합하기도 한다.

상황정보를 위한 저장소로써 제공되는 상황 지식 베이스는 동적으로 상황정보를 하나의 데이터 모델로 연결하여 주고, 다중으로 저장된 상황정보를 추론하기 위하여 상황정보 추론기와 상황정보

질의 엔진을 위한 인터페이스를 제공한다.

상황정보 질의 엔진은 저장된 상황정보와 추론된 높은 수준의 상황정보 질의를 담당하며, 상황정보 추론기에 대한 구조는 4장에서 자세하게 설명하기로 한다.

가상공간에서는 물리적인 공간에서 존재하는 다양한 하드웨어 센서로부터 낮은 수준의 획득한 상황정보와 시맨틱 웹 공간에서 각종 리가시 어플리케이션(날씨 서비스, 뉴스 서비스, 일정관리 어플리케이션)에서 획득한 상황정보를 명확하게 표현하기 위하여 온톨로지기반 상황정보로 모델링 한다.

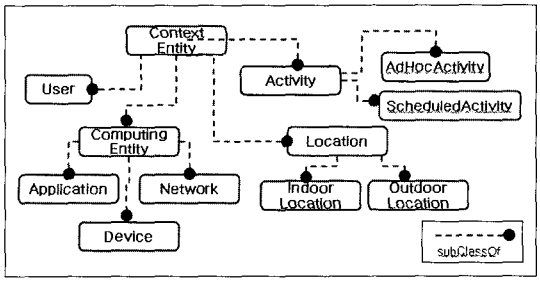
또한 모델링 된 상황정보는 RDF, OWL[8] 같은 표준 시맨틱 웹 기술을 이용하여 구현하였고, 보다 향상된 상황정보 추론, 검증, 질의를 위하여 JENA2 시맨틱 툴킷으로부터 로직을 추론하고, 시맨틱 질의 엔진을 사용하였다.

4. 상황정보 모델링

가상공간에서 온톨로지 기반 상황정보 모델링은 유비쿼터스 컴퓨팅 환경에서 중요한 부분을 차지하고 있고, 모델링이 잘된 상황정보 모델은 상황정보를 추론하여 높은 수준의 상황정보로 변환하여 사용할 수 있다. 따라서 온톨로지 기반 상황정보 모델링은 가상공간에서 상황정보를 공유하고, 시맨틱 질의와 상황정보의 재사용이 가능하게 하고, 추론과 검증을 통하여 얻어진 높은 수준의 상황정보를 지식베이스에 저장하여 필요시 재사용한다[11].

4.1 상위 단계의 상황정보 온톨로지(Upper level Context Ontology)

지식을 표현하고자 하는 해당 도메인에서 온톨로지는 흔히 엔티티(Entity)들의 집합, 엔티티와 엔티티 간의 관계, 인스턴스(Instance), 기능(Function) 등으로 기술한다. 시맨틱 웹 표준인



(그림 2) 상위단계의 상황정보 온톨로지 정의

OWL은 공통 어휘를 기본으로 하여 컴퓨팅 환경에서 해당 도메인의 온톨로지를 정의하여 엔티티를 교환하고 정보를 처리하기 위하여 실증을 들어 설명하고 있다.

물리적인 환경인 스마트 공간에서 작동하고 있는 다양한 센서정보를 가상공간에서 공통적이고 기본적인 온톨로지로 제공하기 위하여 OWL기반 상위 단계 상황정보 온톨로지(ULCO, Upper level Context Ontology)를 정의 하였다.

물리적인 스마트 공간에서 다양하게 존재하는 각종 센서 상황정보를 온톨로지로 모델링 하기 위하여 실 세계 객체인 사용자(User), 위치(Location), 컴퓨팅 자원(Computing Entity)인 3개의 클래스로 정의하고, 개념적인 하나의 활동(Activity) 클래스로 정의하였다.

위에서 정의한 상위 클래스는 하위 단계 수준에서 정의 한 상황정보 온톨로지와 합병(merge)하여 이와 관련된 추가적인 상황정보를 제공한다.

예를 들면 주어진 위치(location)의 상위 단계 상황정보 온톨로지로서 최상위 객체를 모델링 한다면 소음, 날씨, 온도, 사람의 수 같은 상황정보를 얻을 수 있다.

제안한 모델은 상황인식 어플리케이션이 쉽게 이해 할 수 있도록 온톨로지 인스턴스와 속성(Property, 클래스들간의 관계)으로 표현할 수 있다.

4.2 이벤트 상황정보

상황인식 어플리케이션에서 유용하게 사용되는

상황정보는 각종 센서에서 감지되는 정보와 일정, 날씨, 뉴스 등 외부 소프트웨어 들어오는 정보로 분류한다. 여러 종류의 중요한 상황정보는 상황인식 시스템에서 일어나는 이벤트와 밀접하게 관련이 있다. 어떤 한곳의 센서에서 이벤트가 일어날 때 동시에 다른 센서에서 이벤트가 일어나기도 하고, 동시에 일어나는 이벤트를 조합하여 새로운 이벤트를 생성할 수 있다.

각종 센서에서 일어나는 이러한 상황정보의 형태를 지원하기 위한 상황인식 시스템은 가상 공간에서 Event driven 분산 모델을 기반으로 하여 상황정보를 가공/추론하여 서비스를 제공하는 것을 제안한다. 다양한 이벤트는 두 개의 타입으로 정의 하는데 (1) 원시 이벤트(Primitive Event)와 (2) 합성 이벤트(Composite Event)로 분류한다.

상황인식 시스템의 경우 원시 이벤트는 시스템에 내장된 센서나 다른 메커니즘에 의해서 직접적으로 탐지된 낮은 수준의 이벤트이며 , 원시 이벤트의 예는 다음과 같다.

- 감지된 이벤트 : RFID 태그를 가지고 물리적인 공간으로에 들어올 때 탐지된 이벤트
- 일시적인(temporal) 이벤트 : 이벤트가 일어난 후 곧 이어 이벤트가 일어 나는 이벤트
- 소프트웨어 이벤트 : 외부 시스템으로부터 정보를 검색할 때 일어나는 이벤트
- 네트워크 이벤트 : 다양한 센서 랩퍼로부터 존재하는 컴퓨팅 엔티티까지 알려주는 알림 이벤트

합성 이벤트는 원시 이벤트가 단독으로 처리하는 것이 아니라 여러 개의 원시 이벤트가 조합되어 새롭게 정의된 이벤트이고, 상황 발생은 센서에서 감지된 이벤트(Event)에 의해 이루어지고, 기본적으로 제공되는 주요 Event는 (표 1)과 같다.

(표 1) 주요 이벤트 상황 정보

Event Name	Description
Tag Entered	• 태그가 특정영역에 들어옴 • tag: 탐지된 태그 식별자 (eg. "234567") • container: 센서 service container 식별자(eg. "129.254.164.255:2986") • sensor: 탐지 센서 식별자 (eg. "/ETRI/7동/L89") • location: 센서 설치 장소 이름 (eg. "/ETRI/7동/L89") [timestamp]: 이벤트 생성 시각
TagLeft	• 태그가 특정 영역에서 나감 • tag: 태그 식별자 (eg. "234567") • container: 센서 service container 식별자 (eg. "129.254.164.255:2986") • sensor : 탐지 센서 식별자 (eg. "/ETRI/7동/L89") • location: 센서 설치 장소 이름 (eg. "/ETRI/7동/L89") [timestamp]: 이벤트 생성 시각
User Entered	• 사용자가 특정 장소에 들어옴 • user: 사용자 식별자 (eg. "kimhs") • location: 대상 장소 이름 (eg. "/ETRI/7동/L89") [timestamp]: 이벤트 생성 시각
UserLeft	• 사용자가 특정 장소에서 나감 • user: 사용자 식별자 (eg. "kimhs") • location: 대상 장소 이름 (eg. "/ETRI/7동/L89") [timestamp]: 이벤트 생성 시각
Speech Received	• 음성 명령 발생 • speech: 음성 명령 text (eg. "다음") • symbol: 음성 명령 대응 심볼 • container: 센서 service container 식별자 (eg. "129.254.164.255:2986") • sensor: 음성 명령을 받은 센서 식별자 • [user]: 음성 발생 사용자 식별자 eg. "kimhs") • [task]: 명령을 받은 대상 태스크 식별자 [timestamp]: 이벤트 생성 시각
Timer Expired	• task: 알림을 요청한 대상 태스크 식별자 • token: 알림 구분자

4.3 하위 단계의 상황정보 온톨로지(Lower level Context Ontology)

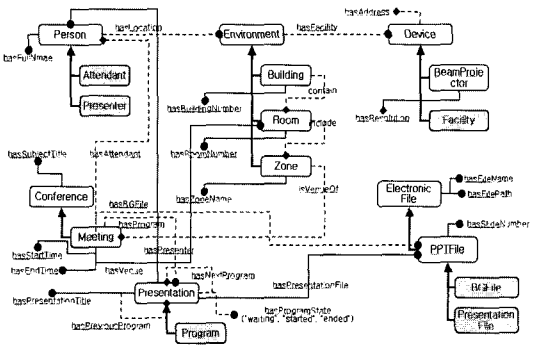
스마트 공간에서 일어난 Event Driven 상황정

보를 앞에서 정의한 상위 단계의 상황정보 모델에 통합하기 위하여 하위 단계의 상황정보 모델을 온톨로지로 확장하여 정의한다.

하위 단계의 상황정보 온톨로지를 정의하기 위하여 물리적인 공간인 스마트 회의장에서 지원하는 간단한 발표 도우미 시나리오를 예를 들어 설명하기로 한다.

- 발표 도우미가 시작하여 발표자가 발표장소에 들어오면 발표장에 있는 전동 스크린이 자동으로 내려오고 실내조명이 꺼진다.
- 발표자가 발표대 앞에 서면 노트북에 있는 발표자의 발표 슬라이드를 띄운다.
- 발표자가 "다음" 명령을 내리면 다음 슬라이드로 이동한다.
- 발표자가 "이전" 명령을 내리면 이전 슬라이드로 이동한다.
- 발표자가 "발표 끝"을 말하면 발표자의 슬라이드 쇼를 끝낸다.
- 자동으로 전동 스크린이 자동으로 올라가면서 조명이 켜진다.
- 대기하고 있던 다음 발표자가 발표장으로 들어오면 이 과정이 반복된다.

위의 시나리오에서 일어날 수 있는 모든 상황을 하위 수준의 상황정보 온톨로지로 모델링 한 결과는 (그림 3)와 같다[11].



(그림 3) 하위단계의 상황정보 온톨로지 정의

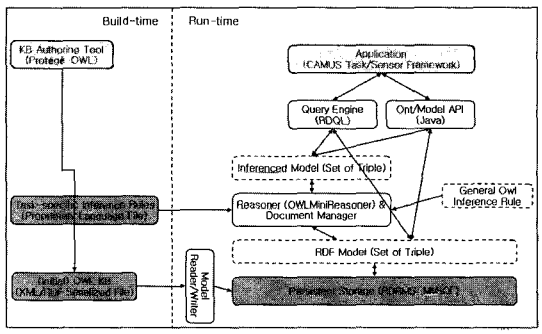
(그림 3)에서 보면 "Environment" 클래스는 "Person" 클래스의 서브 클래스로, "Device" 클래스는 "Environment" 서브 클래스로 모델링 되었고, 그것에 의한 속성(property)관계는 "hasLocation", "hasfacility" 같은 이벤트로 상속되고 있다.

상위 단계의 상황정보 온톨로지 모델은 하위 단계의 상황정보 모델과 합병하여 다양한 상황정보 온톨로지 객체를 상황정보 온톨로지 다른 객체에 추가적으로 연결하여 확장 할 수 있다. 예를 들면 앞에서 정의한 상위 단계 상황정보인 User Class는 ObjectProperty인 HasPerson으로 Person class에 합병하여, 확장된 상황정보 모델을 만들 수 있다.

4.4 상황정보 추론기(Context Reasoner)

상황정보 추론기는 상황 정보를 추론하여 낮은 수준의 상황정보를 높은 수준의 상황정보로 추론하는 컴포넌트이며, 상황정보 수집기로부터 이벤트를 통해 전달된 상황 정보와 이 정보를 바탕으로 암묵적인 지식을 추론하여 지식 베이스에 저장하고 관리한다. 이 계층에서 관리하는 상황 지식은 추후 상황인식 어플리케이션이 실행될 때 참조된다.

따라서, 상황정보 추론기는 상황 지식을 표현을 위한 상황 모델과 함께 상황정보 수집기가 상황 정보를 추가하거나 수정하는 기능, 상황 지식 검색 기능, 그리고 암묵적인 지식의 추론 기능을 제공한다.



(그림 4) JENA2 기반 상황정보 추론기 구조

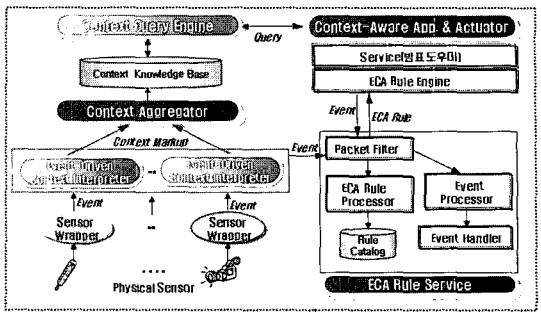
위의 (그림 4)는 (그림 1)의 상황정보 추론기(Context Reasoner)에서 상황 지식을 관리하고 활용하는 동작구조를 보여주며, 개발 시점(Build Time)과 동작 시점(Run Time)을 구분하였다. 우선, 개발 시점에 다양한 기반 상황 지식 저작도구를 이용하여 초기 상황 지식 베이스를 작성하고, 응용에 특화된 추론 규칙을 작성한다.

동작 시점에 초기 상황모델은 (그림 1)의 상황정보 추론기가 관리하는 (그림 4)의 영속적인 저장소(Persistent Storage)에 로딩된다. 이렇게 로딩된 상황모델은 외부에는 RDF 모델과 동일한 개념적 뷰를 통해 제공된다. 이후 상황정보 추론기는 추론 엔진을 적용하여 추론된 모델을 생성된다. 추론된 모델도 외부에는 RDF 모델 뷰로 제공되지만 상황모델에 명시적으로 표현되지 않은 암묵적인 지식들이 유도되어 포함된다는 것이 차이점이다. 추론 엔진이 적용될 때 개발 시점에 작성된 응용에 특화된 사용자 정의 추론 규칙이 추론 엔진 자체의 추론 규칙과 함께 적용된다. 이후 상황 기반 응용들은 검색 엔진을 통해 필요한 지식을 참조하고, 상황모델에 접근하는 기능을 제공하는 API를 이용하여 상황모델에 지식을 생성하거나 변경한다.

5. 구 현

본 장에서는 위에서 정의한 가상공간에서 event driven기반 상황정보를 모델링 한 것을 실험하기 위하여 발표도우미 프로토타입을 구현하였다.

상황인식 시스템 프로토타입에서는 상황인식 어플리케이션과 ECA 룰 서비스로 구성된다. 상황인식 어플리케이션은 개별 액츄에이터(actuator)를 기동시키고 수행중인 프로세스를 관리 또는 제어하는 역할을 한다. 액츄에이터(actuator)는 다시 다수의 작업 규칙 (Task Rule)으로 구성된다. ECA Rule 서비스에서 각 작업 규칙은 ECA (Even Condition Action)의 표현으로, 이벤트 (even)와 조건 (condition)을 통해 환경 및 상황의 변화와



(그림 5) Event Driven 상황인식 시스템 프로토타입

사용자 또는 응용의 요구를 기술하고 해당 상황이 발생될 때 수행할 서비스를 행동 (action)으로 기술한다. ECA 작업 규칙이 수행될 때 필요한 상황 지식을 상황정보 추론기(Context reasoner)를 참조한다.

또한 가상공간상의 상황인식 하부구조에서 제공하는 컴포넌트인 상황인식 엔진, 상황정보 지식베이스, 상황정보 수집기, 정의한 이벤트 인터페이스를 이용하여 구현하였다.

상황정보 모델에서 이벤트에 관한 상황정보인 하위 단계 이벤트 온톨로지는 상황인식 엔진에 의해서 질의를 통하여 검색할 수 있다. 예를 들면 RFID 태그를 가지고 발표장에 들어오면 그 시점에서 일어나는 모든 이벤트를 찾기 위해 RDQL로 된 다음과 같은 질의 엔진으로 다음과 같은 질의를 할 수 있고, 상황정보를 XML로 표현할 수 있다.

(표 2) 일어난 이벤트를 찾기 위한 RDQL로 질의

```
SELECT  ?meeting, ?room
        WHERE
(?commander,<camus:hasLocation>, ?room),
        (?room, <camus:isVenueOf>, ?meeting),
        (?meeting,<camus:hasStartTime>,
?meeting_start_time),
        (?commander, <camus:isAttendantOf>, ?meeting),
        (?meeting,<camus:hasMeetingState>,
        ?meeting_state)
AND     ?meeting_start_time eq \"11:00\"^^xsd:string
        && ?meeting_state eq \"waiting\"^^xsd:string
USING   camusFORhttp://etri.re.kr/sobot/context
        manager#
```

(표 3) RDQL로 질의된 결과

```
<context>
    <nodes>
        <Location id="ETRI/L89 호"/>
        <User id="_김형선"/>
<valued id="_002" class="java.lang.String">회의장
</valued>
    </nodes>
    <associations>
        <association name="light" from="ETRI/L89"
to="_001"/>
        <association name="Meeting" from="L89"
to="_002"/>
    </associations>
</context>
```

ECA 룰 엔진은 ECA 룰 서비스를 호출하여 일어나는 이벤트 상황정보를 필터링하여 룰 카타로그에 업데이트 하고, ECA 룰 서비스는 event driven 상황인식 어플리케이션에게 ECA 룰 관련 정보를 제공한다. 앞에서 정의한 발표도우미 시나리오에서 RFID 태그를 가지고 발표장에 들어오면 전등을 켜는 것을 ECA 룰로 표현하면 다음과 같다.

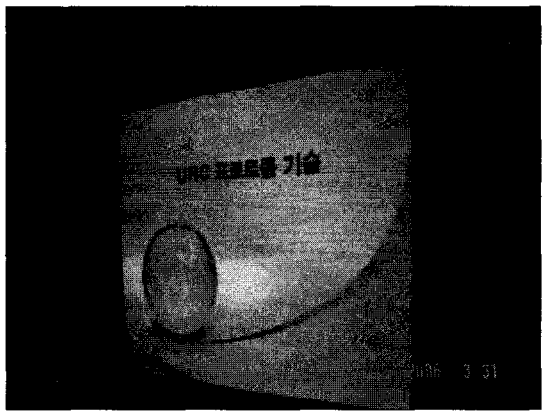
(표 4) 일어난 이벤트를 ECA Rule로 표현

```
on event UserEntered(e)
condition e.location == 'ETRI/7동/L89회의실'
{action e.location.light.turn_on(); }
```

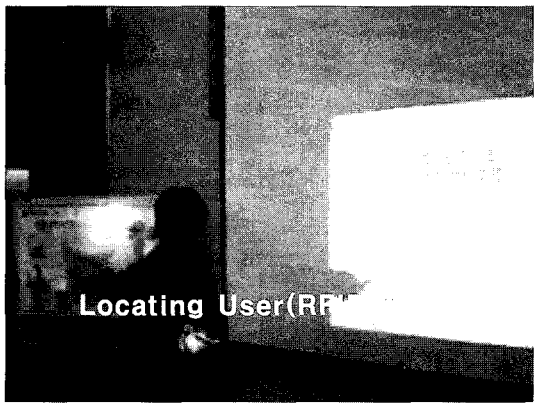
on" 節에는 작업 규칙의 수행을 유발하는 이벤트 연산식(event description)이 기술되고, "condition" 절에서는 해당 작업 규칙이 수행되기 위한 조건을 기술하고, condition절 내부는 "action"에 해당하는 부분으로써, 명시된 조건이 만족하는 경우 수행될 서비스들을 기술한다.

(그림 6)은 발표도우미의 시나리오를 실험한 화면을 나타낸다. 사용자 위치는 RFID 센서와 카메라 센서를 통해서 인식한다. (그림 6-1)은 발표자가 Tag를 가지고 발표장에 들어오면 전동 스크린이 자동으로 내려오고, (그림 6-2)는 전동 스크린이 완전히 내려오고 난 뒤 노트북에 있는 발표

(그림 6-1) 발표자가 발표장에 들어온 경우



(그림 6-2) 노트북의 발표자료 자동 로딩



(그림 6-3) 발표준비가 완료 된 경우

(그림 6-4) "다음" 음성 명령을 내린 경우

자료가 자동으로 로딩되어 전동 스크린에 보여주는 모습이다. (그림6-3)은 발표자료가 완전히 로딩되고 난 뒤 임시로 설치한 판넬의 전등이 두 개 중 1개가 꺼지고 발표준비가 완전하게 준비 된 모습이고, (그림6-4)는 발표자가 음성으로 "다음" 명령을 내렸을 때 다음 페이지로 이동 하면서 발표자가 발표를 진행하는 모습이다.

6. 결 론

물리적 스마트 공간에서 상황정보의 명백한 표현, 의미 있는 질의, 상황정보의 유연한 추론을 위하여 시맨틱 웹 기술을 이용하여 온톨로지 기반 상황인식 구조인 가상 공간을 제시하고, 사용자에게 필요한 서비스를 구현하기 위하여 상황인식 어플리케이션을 제안 하였다. 온톨로지 기반 상황인식 하부구조인 가상공간은 물리공간인 스마트 공간의 여러가지 다양한 센서로부터 상황정보를 수집하기 위하여 위치, 환경, 디바이스 컴포넌트가 서로 협업하는 상황정보 센서 랩퍼(Contex Sensor Wrapper), 상황정보 수집기(Context Aggregator), 상황정보 지식베이스(Context Knowledge Base), 상황정보 추론기(Context Reasoner)와 상황정보 질의엔진(Context Query Engine)으로 구성되어 있다.

본 논문에서는 이러한 컴포넌트를 근간으로 하여 물리적인 스마트 공간에서 일어나는 Event Driven 상황정보를 가상공간으로 맵핑하기 위하여 온톨로지 기반 상황인식 프레임워크를 제안 하였고, 가상공간으로 들어온 Event driven 기반 하위 수준의 상황정보를 상위수준의 상황정보와 맵핑하기 위하여 온톨로지 기반 상황정보 모델로 모델링 하는 방법과 상황인식 시스템을 구현 하였다.

모델링 방법은 스마트 공간에서 공통으로 사용할 수 있는 공통 온톨로지(Common ontology)를 상위 단계의 온톨로지로 모델링 한 다음, 이벤트가 일어날 때 마다 발생하는 상황정보만을 필터링하여 하위단계의 상황정보 온톨로지로 모델링 한다.

따라서 기존 연구에서는 상위단계의 온톨로지(Common ontology)와 하위 단계의 이벤트 온톨로지로 구분하지 않고 모든 상황정보를 통합하여 모델링하기 때문에 추후에 재사용이 불가능하고, 또한 이렇게 모델링된 상위단계의 상황정보와 하위단계의 상황정보를 연결하기 위하여 Merge(맵핑)하는 방법을 제안한 것이 본 논문에서의 특징이라고 볼 수 있다. 또한 맵핑된 High level 상황정보는 가상공간에서 ECA(Event, Condition, Action) 룰에 따라 사용자에게 언제 어디선지 사용자가 필요로 하는 능동적이고 지능적인 서비스를 제공하고, 스마트 오피스 환경에서 발표 도우미 서비스를 구현하고, 간단한 실험을 수행 하였다.

참 고 문 헌

[1] A. Dey, D. Abowd and D. Salber, "Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context Aware Applications," Human Computer Interaction (HCI) Journal, Vol. 16, 2001

[2] H. Kim et al., "A Software Robot Technology in URC," Journal of The Korean Institute of Communication Sciences, Vol. 21, No. 10, 2004, pp. 36-43

[3] R. Want, A. Hopper, V. Falcao and J. Gibbons. "The Active Badge Location System," ACM Transactions on Information Systems, Vol. 10, No. 1, 1992, pp. 91-102

[4] G. Abowd et al., "Cyberguide: A Mobile Context aware Tour Guide", Wireless Networks, Vol. 3, No. 5, 1997, pp. 421-433

[5] A. Dey, M. Futakawa, D. Salber, and G. Abowd, "The Conference Assistant: Combining Context Awareness with Wearable Computing," Proc. Symp. Wearable Computers, 1999, pp. 21-28

[6] P. Brown. "Triggering Information by Context", Personal Technologies, Vol. 2, No. 1, 1998

[7] B. Schilit, N. Adams, and R. Want. "Context aware Computing Applications," Proc. IEEE Workshop on Mobile Computing Systems and Applications, 1994, pp. 85-90

[8] Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel Schneider, Lynn Andrea Stein, and Sean Bechhofer, "OWL Web Ontology Language Reference", World Wide Web Consortium (W3C), February 2004.

[9] Anid K. Dey and Gregory D. Abowd. "Towards a Better Understanding of Context and Context awareness", Technical Report GIT GVU 99 22, Gorgia Institute of Technology, College of Computing june, 1999.

[10] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Koooer, and Mike Pinkerton, "Cyberguide: A Mobile ContextAware Tour Guide, "Wireless Networks", Vol.3, No5, Oct. pp.421 433,1998

[11] Joonmyun Cho, Hyun Kim, Hyungsun Kim, Joohaeng Lee, Choongsung Hong, Jinmi Jung, "Context Knowledge Modeling for Ubiquitous Computing", The 2nd International Conference on Ubiquitous Robots and Ambient Intelligence (URAmI2005), pp.283, Daejeon, 2005

[12] Herry Chen, Tim Finin, and Anupam Joshi, "An ontology for context aware pervasive computing environments", The Knowledge Engineering Review 18(3), 2003.

[13] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung, "Ontology based context modeling and reasoning using OWL", Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.

[14] Jena - A Semantic Web Framework for Java, http://jena.sourceforge.net/, Hewlett Packard Development Company, LP

[15] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel Schneider, editors. The Description Logic Handbook. Cambridge University Press, 2003.

[16] A.K. Dey, Gregory D. Abowd, and Daniel Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of contextaware applications", Human Computer Interaction 16(2), 2001.

[17] K. Henricksen, J. Indulska, and A. Rakotonirainy, "Modeling context information in pervasive computing systems", Proceedings of the First International Conference on Pervasive Computing, volume 2414 of Lecture Notes in Computer Science, Zurich, August 2002.

[18] Anand Ranganathan and Roy H. Campbell, "A middleware for context aware agents in ubiquitous computing environments", ACM/ IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil, 2003.

[19] Herry Chen, Tim Finin, and Anupam Joshi, "An ontology for context aware pervasive computing environments", The Knowledge Engineering Review 18(3), 2003.

[20] 정진미, 이강우, 김현, "URC에서의 상황인식 컴퓨팅, ETRI 전자통신동향분석, pp.33 42, 2005

◐ 저 자 소 개 ◑

김 형 선
1981년 상지대학교 경영학과 졸업(학사)
1990년 광운대학교 대학원 전자계산학과 졸업(석사)
2002년 대전대학교 대학원 컴퓨터공학과 졸업(박사)
1985년~현재 한국전자통신연구원 소프트웨어로봇연구팀 책임연구원
관심분야 : 상황인식 시스템, 분산컴퓨팅. 분산데이터베이스, 전자상거래
E-mail : kimhs@etri.re.kr

김 현
1984년 한양대학교 기계설계학과 졸업(학사)
1987년 한양대학교 대학원 기계설계학과 졸업(석사)
1997년 한양대학교 대학원 기계설계학과 졸업(박사)
1990년~현재 한국전자통신연구원 소프트웨어로봇연구팀장, 책임연구원
관심분야 : 상황인식 시스템, 유비쿼터스 컴퓨팅, 지식공학, 분산협업 설계
E-mail : akmoon@etri.re.kr

문 애 경
1992년 영남대학교 전산공학과 졸업(학사)
1996년 영남대학교 대학원 컴퓨터공학과 졸업(석사)
2000년 영남대학교 대학원 컴퓨터공학과 졸업(박사)
2000년~현재 한국전자통신연구원 소프트웨어로봇연구팀 선임연구원
관심분야 : 분산/병렬 데이터베이스. 상황인식 시스템
E-mail : akmoon@etri.re.kr

조 준 면
1993년 한국과학기술원 기계공학과 졸업(학사)
1995년 한국과학기술원 자동화및설계공학과 졸업(석사)
2006년 한국과학기술원 기계공학과 졸업(박사)
1995~2001년 (주)볼보건설기계코리아 기술연구소 (구 삼성중공업 중앙연구소) 선임연구원
2003~현재 한국전자통신연구원 소프트웨어로봇연구팀 선임연구원
관심분야 : Ontological Engineering, Pervasive Computing, 설계제조 정보 시스템.
E-mail : jmcho@etri.re.kr

홍 충 성
1998년 홍익대학교 산업공학과 졸업(학사)
2000년 홍익대학교 대학원 정보산업공학과 졸업(석사)
2003년 홍익대학교 대학원 정보산업공학과 수료(박사)
2003년~현재 한국전자통신연구원 소프트웨어로봇연구팀 연구원
관심분야 : 유비쿼터스 컴퓨팅, 상황 정보 모델링
E-mail : cshong@etri.re.kr