

온톨로지 기반의 컨텍스트 정보 모델링 기법

김진형*, 황명권*, 정한민*

요약

유비쿼터스 컴퓨팅 환경에서는 주변 환경에 의해 발생하는 방대한 컨텍스트 정보에 대한 모델 정의와 컨텍스트 인지를 통한 지능적인 서비스 제공을 위하여 컨텍스트 정보 처리, 관리 및 추론과 관련한 연구가 필수적으로 요구된다. 그러나 현재 컨텍스트 인지 분야에서는 고수준의 서비스 지원을 위한 온톨로지 특성을 효과적으로 반영한 모델링 기법이 부재하며, 지능적인 추론(응용, 조합)을 지원하는 기법 부재 및 컨텍스트 정보간 상호운용성 지원의 한계를 가지고 있다. 따라서 본 연구의 최종 목표는 실제 세계에 존재하는 실체(entity)에 대한 상태를 특징화하고 정의하기 위한 방법으로 육하원칙을 적용한 온톨로지 기반의 컨텍스트 인지 모델링 기법 컨텍스트 인지를 위한 육하원칙 온톨로지를 개발하고 보다 양질의 지능화된 컨텍스트 인지 서비스를 제공하기 위해 컨텍스트 정보에 대한 관리 및 컨텍스트 추론을 지원하는 프레임워크를 개발함에 있다.

Context Information Modeling Method based on Ontology

Jinhyung Kim*, Myungwon Hwang*, Hanmin Jung*

Abstract

Ubiquitous Computing is required to define models for broad context information occurrence by surrounding environment and to study how to model a mechanism for selectively collecting useful pieces of context information and providing relevant intelligent services. Further, studies are also required as to process of context information, and its maintenance and reasoning. However, current context-aware research area still lacks modeling technique that reflects the characteristics of ontology effectively for providing relevant intelligent services. It has also limitation about context reasoning and interoperability among context information. Therefore, this paper proposes ontology-based context-aware modeling technique and framework enabling efficient specification of context information for providing intelligent context-aware services that support context management and reasoning.

Keywords : Context Information Modeling, Context Awareness, Ontology Modeling

1. 서론

유비쿼터스 컴퓨팅은 다양한 디바이스가 현실 세계에 사물과 환경에 스며들어 서로간의 상호작용을 통해 언제 어디서나 이용할 수 있는 인간, 사물, 정보간의 컴퓨팅이 가능한 환경으로서 이러한 컴퓨팅 환경을 구현하기 위해서는 다양

한 시스템이 환경, 시간, 장소, 사용자 등에 대한 컨텍스트 인지 기술을 지원해야 한다. 컨텍스트 인지 기술은 컨텍스트의 변화를 감지하고 사용자/단말기에 적합한 정보나 서비스를 제공하거나 시스템이 스스로 상태를 변경하는 것이 핵심이다[1]. 이러한 컨텍스트 인식 기술을 지원하기 위해서는 센서로부터 받은 컨텍스트 정보를 이용하여 비가시적이고 자동화된 처리가 가능한 환경 및 기술을 제공해야 한다. 즉, 컨텍스트의 변화를 감지하고 컴퓨터 스스로 인식/처리하기 위한 보다 지능적인 컨텍스트 인지 모델에 대한 연구가 필요하다. 기존에 제안된 유비쿼터스 컴퓨팅 환경에서의 컨텍스트 인지 기술은 컨텍스트에 맞는 최적의 서비스를 제공하기 위해

※ 제일저자(First Author) : 김진형
접수일:2011년 09월 23일, 수정일:2011년 11월 02일
완료일:2011년 12월 06일
* 한국과학기술정보연구원 정보기술연구실
{jinhyung, mgh, jhm}@kisti.re.kr

Agent를 이용하여 컨텍스트 정보를 수집 및 처리하였다[2-14]. 또한, 최근에는 지능화 된 센서와 디바이스, 시스템을 기반으로 센서 네트워크, 컨텍스트 인지 기술이 발전함에 따라 온톨로지를 활용한 컨텍스트 정보 간의 상호운용성 및 의미 해석을 기계가 스스로 처리할 수 있는 컨텍스트 정보 추론 지원 등에 대한 연구가 활발히 진행되고 있다[15-18]. 그러나, 현재 유비쿼터스 컴퓨팅 환경에서 온톨로지를 활용한 지능적인 컨텍스트 인지 처리를 위한 많은 연구가 제안되고 있음에도 불구하고 아래와 같은 여러 가지 이슈를 가진다.

첫째, 온톨로지 기반의 컨텍스트 인지 시스템 개발에 있어 컨텍스트 인지에 대한 모델링 수준이다. 기존의 컨텍스트 인지 시스템은 하나의 도메인에 특화되어 개발된 강결합 컨텍스트 인지 모델과 범용적인 약결합 컨텍스트 인지 모델로 구분 가능하다. 강결합 컨텍스트 인지 모델은 특정 도메인에 대해 고성능의 서비스를 제공할 수 있다는 장점을 가지는데 반해 타 도메인의 컨텍스트 처리를 위해서는 부가적인 컨텍스트 처리 방법이 요구된다. 또한 약결합 컨텍스트 인지 모델은 다양한 컨텍스트 정보 응용에서 범용적인 활용이 가능한 장점을 가지지만 컨텍스트 정보에 대한 컨텍스트 응용과 조합 등 추론을 요구하거나 고성능의 서비스 제공은 불가하다.

둘째, 다양한 디바이스와 센서 등에서 생성된 컨텍스트 정보에 대한 상호운용성이다. 다양한 디바이스와 센서들에서 수집된 컨텍스트 인지 정보는 다양한 모델을 통해 정의되고 있다. 하지만 각 모델의 의미 정의 방법이 다양하고 상이함에 따라 각 컨텍스트 인지 정보 간의 비교 및 분석을 위해서는 많은 비용이 소요되게 된다. 따라서 다수의 소스로부터의 수집되는 다양한 형태의 컨텍스트 인지 정보들에 대한 상호운용성이 유지되어야 좀 더 지능적인 컨텍스트 인지 시스템을 구현할 수 있다.

마지막으로, 컨텍스트 응용과 컨텍스트 조합을 위한 컨텍스트 추론을 아직 완벽하게 지원하지 못한다. 특히, 컨텍스트 인지 기술에서의 컨텍스트 응용(A 컨텍스트를 통해 B컨텍스트가 발생된 서비스)과 컨텍스트 조합(A컨텍스트하에서 B컨텍스트 + C컨텍스트 조합 = D컨텍스트)에 대한 처리는 유비쿼터스 컴퓨팅 환경에서 지

능화된 맞춤형 서비스를 지원하기 위해서는 반드시 해결해야 하는 이슈사항이다.

이러한 문제점을 해결하기 위하여 유비쿼터스 컴퓨팅 환경에 존재하는 방대한 컨텍스트 정보의 추상화 정도에 대한 효율적 모델 정의와 유용한 컨텍스트 정보를 선택적으로 수집하고 컨텍스트에 맞는 지능적인 서비스 제공을 위한 모델링 기법과 컨텍스트 정보 처리, 관리 및 추론에 대한 연구가 필요하다. 따라서, 본 논문에서는 유비쿼터스 서비스 환경에서 효율적인 컨텍스트 정보 처리를 지원하기 위한 온톨로지 기반의 컨텍스트 인지 모델링 기법을 제안한다. 논문에서 제안하는 컨텍스트 인지 모델링 기법을 지원하기 위해 물리적, 논리적 컨텍스트 정보에 대한 육하원칙(what, where, how, when, why, how)을 활용하여 컨텍스트 정보를 모델링 한다. 육하원칙을 적용한 컨텍스트 인지 모델링은 직관적인 컨텍스트 인지 스키마를 제공함으로써 컨텍스트 정보에 대한 높은 상호운용성과 활용도를 보인다. 또한, 제안 모델링은 컨텍스트 정보에 대한 추론을 지원 함으로서 보다 지능적이고 신뢰성 높은 서비스를 제공한다.

2. 관련 연구

2.1 컨텍스트 정보 모델링 기법

현재까지 컨텍스트 정보를 모델링 하기 위해 많은 기법 및 모델들이 제안되었으며 대표적인 컨텍스트 정보 모델링 기법들은 다음과 같다. Key-Value models [2-4]은 컨텍스트 정보를 모델링하기 위한 가장 간단한 데이터 구조이다. 환경변수로서 컨텍스트 정보의 수치를 어플리케이션에게 제공하는데, 키-값 쌍의 형식을 사용하여 컨텍스트를 모델링한다. Key-Value 모델은 관리하기 편하다는 장점을 가지고 있지만, 효과적인 컨텍스트 검색 알고리즘을 위한 복잡한 구조화는 부족하다는 단점을 가지고 있는 모델링 기법이다. Markup scheme models [5-7]은 속성과 내용을 갖는 마크업 태그로 구성된 계층적 데이터 구조이다. 마크업 태그의 내용은 항상 다른 마크업 태그에 의해 재귀적으로 정의된다. 이러한 종류의 컨텍스트 모델링 기법에서는 프로파일들이 대표적이다. 프로파일은 항상 XML과 같은

모든 마크업 언어의 슈퍼클래스인 SGML에서 유도된 직렬화를 기반으로 한다. Object-oriented models [8-10]은 유비쿼터스 환경에서 컨텍스트의 동적 특성에 관한 문제점들을 해결하기 위하여 캡슐화와 재사용이라는 객체 지향의 주요 장점을 적용한 기법이다. 컨텍스트 처리에 관한 자세한 내용은 개체 수준에서 캡슐화되어 다른 컴포넌트들에게 은닉되고 컨텍스트 정보로의 접근은 특정인터페이스를 통해서만 제공된다. Logic-based models [11-14]은 규칙을 기반으로 하여 다양한 컨텍스트에 대한 지식을 룰로 표현하고 컨텍스트에 따른 값을 어플리케이션에 제공한다. 그러나 논리기반 모델링은 규칙이 많아질수록 관리가 어렵고 규칙간의 충돌해소 해결에 많은 시간과 비용이 소요된다는 단점이 있다. Ontology-based models [15-18]에서 온톨로지는 개념과 상관 관계를 기술하는 도구[19, 20]로 컨텍스트 정보를 표현하고 공유하기 위한 어휘 및 용어를 제공하며 컨텍스트 정보의 다양성 때문에 도메인 기반 온톨로지를 정의한다. 이 모델은 유비쿼터스 컴퓨팅 환경에서 컨텍스트지식의 공유와 재사용을 지원하며, 계층적 컨텍스트 온톨로지 모델에서는 상위 계층의 온톨로지를 이용하여 도메인에 맞는 하위 계층의 온톨로지를 생성한다. 온톨로지를 기반으로 컨텍스트 인지 모델링을 한다면 컨텍스트 정보들간의 관계성을 바탕으로 의미 해석을 기계가 스스로 처리할 수 있는 컨텍스트 정보 추론을 지원할 수 있다는 장점이 있다.

2.2 온톨로지 기반 컨텍스트 인지 시스템

이 절에서는 기존에 제안된 온톨로지 기술을 적용한 컨텍스트 인지 시스템에 대하여 기술한다.

CoBra는 온톨로지 기술언어 중에서OWL을 지원하는 시스템으로 Intelligent Space라고 불리는 유비쿼터스 환경에서 컨텍스트 정보의 생성/관리/배포를 지원하기 위한 에이전트를 기반으로 한 아키텍처이다. CoBrA에서는 컨텍스트 정보를 온톨로지 모델링을 위하여 CoBrA ontology를 제안하였다. COBRA-ONTO는 시스템에 독립적인 표현 온톨로지 개발을 목표로 하여 OWL로 설계되었다. COBRA-ONTO는 행동, 에이전트, 장치, 회의, 디지털 문서, 공간, 시간에 관한 온

톨로지를 정의하여 센서나 태그 값, 또는 사용자의 시간과 장소에 따라 서비스 제공이 가능하도록 추론 기능 및 웹 서비스로의 전송을 위한 컨텍스트관단을 지원한다. 그러나 CoBrA의 경우, 발생하는 모든 Context를 공유 및 관리하기 위한 브로커 에이전트 개발에 초점을 두어 온톨로지를 기반으로 한 컨텍스트 인지 모델링 부분에서 직관적이지 않은 온톨로지 특성을 정의하여 사용하고 있어 다른 컨텍스트 인지 온톨로지와의 상호운용성에 한계를 가진다.

SOCAM은 SOA를 기반으로 OSGi위에서 구동되는 미들웨어로서, OWL기반의 온톨로지를 사용함으로써 컨텍스트에 대한 표현과 다양한 형태의 컨텍스트에 대한 추론을 지원한다. 미들웨어 내에서는 컨텍스트 정보를 온톨로지 모델링하기 위하여 CONON을 사용한다. CONON은 여러 도메인에 쉽게 적용할 수 있도록 대부분의 컨텍스트 인지 시스템이 가지는 상위 클래스들을 온톨로지 모델로 정의한다. 이를 통해 다른 도메인 시스템은 상위 클래스(온톨로지)를 가져다가 사용할 수 있으며, 공유 역시 쉬워지는 장점을 가진다. 그러나 SOCAM의 경우 SOA를 기반으로 지식의 공유, 분류와 상호 의존성과 관련된 문제를 해결하는 것을 목적으로 하고 있으나, 컨텍스트 정보 센싱을 위해서 매우 복잡한 접근방법을 이용한다는 단점을 가지고 있으며, CONON의 경우 약결함 형태의 온톨로지 모델링 방법이기 때문에 컨텍스트 정보에 대한 컨텍스트 응용과 조합 등 추론을 요구하거나 고성능의 서비스를 제공하기 위한 측면에서 한계가 있다.

GAIA[17]는 컨텍스트 정보를 나타내기 위해 전통적인 Operating System 개념을 확장한 것이라 볼 수 있다. GAIA는 Ontology서버를 이용하여 객체에 대한 정보를 수집하고 센서 네트워크를 이용하여 컨텍스트 정보를 수집한다. 그리고 수집한 객체 정보 및 컨텍스트 정보를 기반으로 추론하여 상위 Context(Higher level context)를 추출한다. GAIA에서 정의한 객체는 Service, Device, Application, and User 등을 포함하고, 컨텍스트 정보를 나타내기 위해 Location, Activity, Weather Information등을 포함한다. GAIA의 Context 모델은 DAML+OIL을 기반으로 주어 목적어 사이에 서술어 모델을 넣는 방식이지만 서술어 자체에 대한 정의 및 표현은

없기 때문에 의미적으로 완벽하게 컨텍스트 인지를 한다고 보기는 어렵다. 또한 Gaia에서 활용되는 DAML+OIL은 지능적 맞춤형 서비스를 제공하기 위한 추론 기능에 있어서 OWL 보다 기능적인 한계점을 가지게 된다.

Context Toolkit은 객체지향 접근법을 이용하여 센서기반의 컨텍스트 인지 서비스를 용이하게 개발할 수 있는 프레임워크와 재사용이 가능한 모듈을 제공하여 컨텍스트를 쉽게 이용할 수 있게 지원해주는 Toolkit으로 엄밀히 따지자면 Context Toolkit은 온톨로지 기술을 따르지 않지만 온톨로지의 성격을 띠는 Attribute-value의 tuples을 이용하여 모델링 한다. 그러나 본 연구에서는 추론엔진이 존재하지 않기 때문에 컨텍스트 인지의 지능화된 서비스 제공에 한계가 있으며 컨텍스트 정보를 처리하기 위한 프로그래밍이 복잡하다는 단점을 가지고 있다.

위에서 언급한 기존 연구들은 대부분 온톨로지를 기반으로 컨텍스트 인지 모델링을 제안하고 있지만, 단지 여러 컨텍스트 정보를 온톨로지 언어로 정의하는 데만 초점을 두고 있다. 또한, 모델링 기법이 시스템마다 모두 다르다는 점과 모델링된 컨텍스트 인지 온톨로지의 경우 추론과 같은 온톨로지의 특성을 잘 나타내지 못하는 점과 같다. 따라서 정의되지 않은 컨텍스트 정보에 대해 한계를 가지며 서로 다른 방식의 모델링 기법의 사용은 다른 시스템간의 상호운용성 문제와 이들 간 확장 및 통합이 어렵다는 문제점을 야기시킨다. 따라서 본 연구에서는 정의되어 있지 않은 컨텍스트와 온톨로지 개념에 대한 분류와 조합이 가능하도록 온톨로지 기반의 컨텍스트 인지 모델링 기법을 제안한다.

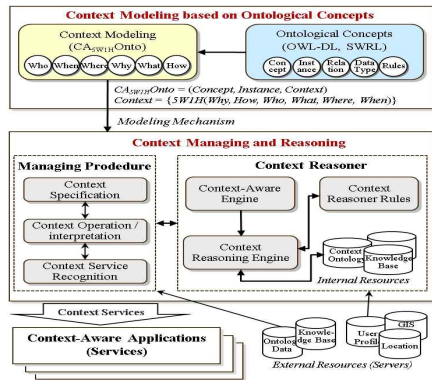
3. 육하원칙 온톨로지 기반의 컨텍스트 인지 모델

Context는 지식의 특정한 형태로 컨텍스트 인지에 대한 모델링에 있어 중요한 고려사항으로 컨텍스트 정보에 대한 expressiveness와 complexity에 대한 tradeoff를 들 수 있다. 이러한 tradeoff는 컨텍스트 처리를 위한 지식(컨텍스트)표현과 추론의 능력을 결정짓는 중요한 요소이다. 컨텍스트 인지 모델링에 대한 이러한

tradeoff를 해결하기 위한 방안으로 온톨로지 (i.e., OWL-DL)를 활용할 수 있다. 온톨로지는 concepts과 그들의(concepts)의 relationships을 나타냄에 있어 풍부한 표현력과 컨텍스트에 대한 처리 시 다양한 표현을 기반으로 자동화된 추론을 지원한다. 따라서, 본 논문에서는 유비쿼터스 환경에 존재하는 컨텍스트 정보에 대한 entity를 특성화하고 실체화 하기 위한 방법으로 온톨로지를 활용한 온톨로지 기반의 컨텍스트 인지 모델링(CA5W1HOnto) 기법을 적용하여 컨텍스트 정보에 대한 효율적인 명세가 가능한 프레임워크를 제안하고, 이를 위한 컨텍스트 처리 프로세스를 정의한다.

3.1 육하원칙 온톨로지 기반의 컨텍스트 인지 시스템

유비쿼터스 환경에서 컨텍스트의 변화를 감지하고 컨텍스트에 맞는 서비스를 제공하기 위해서는 컨텍스트에 대한 변화를 시스템/센서가 실시간으로 감지하고 처리할 수 있어야 한다. 이를 위해서는 컨텍스트 정보에 대한 컨텍스트 인지 모델링 기법 및 컨텍스트 관리와 컨텍스트 추론이 요구된다. 따라서, 본 절에서는 컨텍스트 정보에 대한 정확한 정의 및 관리를 위해 육하원칙을 적용한 온톨로지 기반의 모델링 기법을 정의하고, 이를 활용한 컨텍스트 관리 및 컨텍스트 추론을 지원하는 프레임워크를 제안한다. 본 논문에서 제안하는 컨텍스트 정보 처리를 위한 프레임워크는 의미적인 컨텍스트에 대한 해석과 추상화를 지원하기 위한 방법으로 육하원칙을 적용한 온톨로지 기반의 모델을 정의한다. 제안 모델은 OWL-DL을 통해 컨텍스트 정보에 대한 formalism을 제공하며 컨텍스트 정보에 대한 추론 서비스를 지원한다.



(그림 1) 육하원칙 온톨로지 기반의 시스템 프레임워크

현재까지 컨텍스트 인지와 관련한 컨텍스트 정보 처리에 대한 다양한 모델링 기법들이 연구되었다. 특히 온톨로지를 기반한 컨텍스트 인지 모델은 다음과 같은 장점을 가진다. (1) 온톨로지는 컨텍스트에 대한 간결한 표현으로 풍부한 관계성을 지원한다. (2) 컨텍스트 정보에 a formal semantics 은 서로 다른 자원의 컨텍스트에 대한 공유와 통합을 가능하게 한다; (3) 온톨로지 Tools(e.g., Protégé)은 컨텍스트에 대한 표현 시 일관성 있는 관계성 정의를 가능하게 한다; (4) 다양한 컨텍스트 정보 표현에 대한 일관성 있는 자동화된 추론을 지원한다.

본 논문에서는 이러한 온톨로지의 특성을 반영하여 컨텍스트 정보에 대한 보다 지능화된 처리와 관리를 위한 컨텍스트 정보 모델을 제안한다. 제안 모델은 다양한 도메인에서 사용되는 서로 다른 컨텍스트 인지 스키마간의 직관적인 통합을 지원하기 위한 방안으로 컨텍스트 인지 스키마에 대한 육하원칙을 적용한다. 제안 모델은 Concept, Instance, Context 3개의 컴포넌트로 구성된다. Concepts과 Instances 의 개념은 기존 온톨로지의 기본 특성들을 반영하여 정의하며 Contexts는 컨텍스트 정보의 다양한 activity로 구성되며 육하원칙의 기본속성과 컨텍스트 인지 구성요소 간 사상되어 컨텍스트 인지 스키마 정의에 사용된다. 또한 육하원칙을 적용한 컨텍스트 인지 모델은 직관적인 모델로써 다양한 도메인들에서 사용되는 스키마들 간의 공유 또는 통합 시 시스템/모델 간 상호운용성 지원을 가능하게 한다. 제안 모델에서 정의하는 육하원칙의

속성과 컨텍스트 인지 구성 요소간의 사상 방법 및 컨텍스트 인지 프로세스는 뒤에서 자세히 설명한다.

3.2 육하원칙 온톨로지 기반의 컨텍스트 저오 모델링

서로 다른 도메인에서 사용되는 컨텍스트 인지에 대한 처리는 매우 복잡하기 때문에 다양한 컨텍스트를 고려한 모델이 요구된다. 이러한 복잡하고 다양한 컨텍스트 인지 모델을 개발함에 있어 컨텍스트 정보에 대한 육하원칙의 활용은 컨텍스트 정보에 대한 공유와 통합에서 직관적이며 고수준의 컨텍스트 인지 서비스를 제공할 수 있다.

본 절에서는 육하원칙을 적용한 컨텍스트 인지 모델링 기법에 대해 기술한다. 제안 모델은 triple < Concepts, Instances, Contexts >로 구성되며 Concepts와 Instances components는 기존 온톨로지에서 정의된 특성을 제안 모델에 수용하고, Contexts components는 육하원칙을 구성하는 속성인 why, who, how, what, where, 그리고 when로 구성되며, 컨텍스트 인지를 위한 ingredients로 Role, Goal, Process, Action, Location, 그리고 Time와 같은 6개의 클래스로 정의된 육하원칙의 속성과의 사상관계를 가진다.

그림 2는 CA5W1HOnto을 구성하는 육하원칙의 속성과 컨텍스트 인지 정의에서 필수적으로 요구되는 ingredients간 사상 방법 및 모델을 나타낸다. CA5W1HOnto에서 하나의 Context 를 정의하기 위한 구성요소로서 Role, Goal, Process, Action, Location, 그리고 Time을 가지며 육하원칙과의 사상관계는 why::Goal, who::Role, how::Process, what::Activity, where::Location, 그리고 when::Time으로 나타낸다.

본 논문에서 제안하는 컨텍스트 인지 모델의 why::Goal 은 컨텍스트 인지의 처리의 최종 목표로 육하원칙의 Why에 해당되며, Personal_Goal, Functional_Goal, NonFunctionl_Goal, 그리고 Role_Goal 로 구성된다. who::Role은 why::Goal을 처리하기 위한 역할을 정의한 것으로 Role_Goal, Organization, 그리고 Actor 가 가지는 roles을 정의한다. who::Role은 육하원칙의 who에 해당된다. 또한

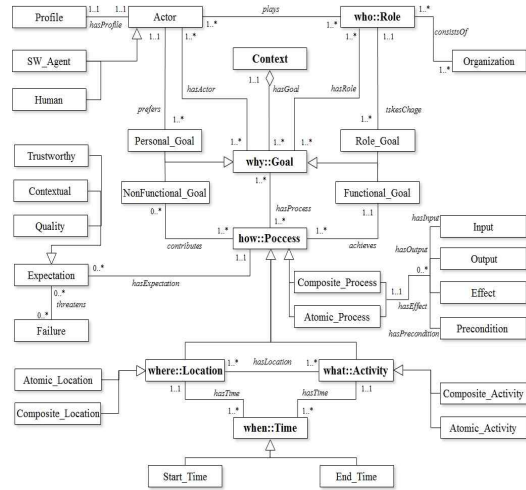
Actor의 역할을 정의하기 위해 subclasses 로서 Profile과 SW_Agen등을 가진다. how::Process는 육하원칙의 how에 해당하는 속성 클래스로서 컨텍스트 인지 서비스의 처리 절차를 나타낸다. how::Process는 why::Goal 이 실현될 수 있도록 컨텍스트 인지의 처리 단계를 나타내며, Expectation, Atomic_Process, 그리고 Composit_Process를 가진다. what::Activity는 컨텍스트 객체가 가지는 센싱이 가능한 고유한 특성으로 육하원칙의 what에 해당되며, Atomic_Activity와 Composit_Activity로 나타낸다. 그리고 where::Location는 육하원칙의 where에 해당하는 클래스로 컨텍스트의 위치와 관련된 정보를 제공하며, 마지막으로 when::Time는 육하원칙의 when 에 해당되는 것으로 Start_Time과 End_Time으로 컨텍스트 인지에 대한 처리를 가능하게 하는 속성 요소로 사용된다.

본 논문에서 제안하는 컨텍스트 인지 모델은 다음과 같이 정의 가능하다.

- OntoCA = {Concepts, Instances, Contexts}
- Context Model = {5W1H(Who, Why, How, What, Where, When,) }
- Who = {Role (actor(profile), organization)}
- Why = {Goal (personal_goal, role_goal, nonfunctional_goal, functional_goal)}
- How = {Process (Expectation_Process(Failure), Input_Process, Output_Process Effect_Process, Precondition_Process)}
- What = {Activity (atomic_action, composite_action)}
- Where = {Location (atomic_location, composite_location)}
- When = {Time (start_time, end_time)}

3.3 컨텍스트 인지 정보 처리 프로세스

본 절에서는 컨텍스트 인지 모델의 컨텍스트 정보 처리 절차에 대해 기술한다. 다양하며 동시다발적으로 발생하는 컨텍스트 정보들을 효율적으로 처리하기 위해서는 Role Layer, Goal Layer, Process Layer, Activity Layer, Location

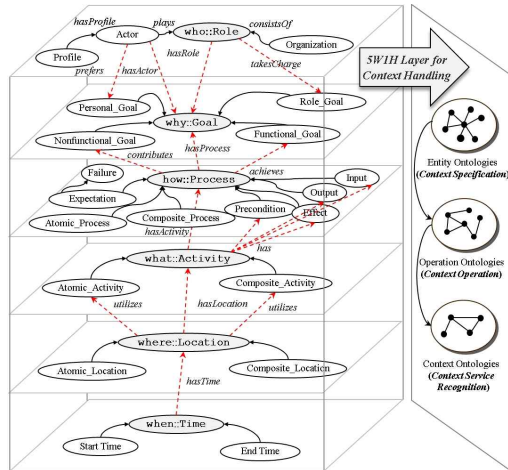


(그림 2) 육하원칙 기반 온톨로지 모델 Layer, 그리고 Time Layer에서 정의한 컨텍스트 정보 처리 절차를 거쳐야 한다. 그림 3은 본 논문에서 제안하는 제안한 컨텍스트 인지 모델의 컨텍스트 인지 정보 처리 프로세스에 대한 나타낸다.

컨텍스트 인지 정보 처리를 위한 첫 번째 프로세스는 객체가 가지는 고유한 특성 중 수집이 가능한 부분이 무엇인지를 선별하는 데이터 선정 단계이다. 이는 육하원칙의 What 에 해당되는 부분으로 컨텍스트 처리 요소의 Activity Layer에서 처리되며 육하원칙의 Where와 When에 해당되는 Location Layer와 Time Layer는 Activity Layer에서 수집된 데이터에 대한 장소와 시간의 정보를 제공한다. 그리고 Process Layer는 수집된 컨텍스트 정보를 어떻게 처리 할 것인지를 담당하는 Layer로 Precondition, Expectation, Effect, Input, 그리고 Output과 같은 속성의 처리 결과를 Goal Layer 에게 넘겨준다. 이때 Constraint에 대한 처리는 Precondition에서 사전 처리한다. 마지막으로, Goal Layer 는 Process Layer와 Role Layer에서 얻은 처리 결과를 통해 컨텍스트에 대한 처리/정의를 위한 서비스를 선정한다.

이렇게 6단계의 Layers을 통해 모델링 된 컨텍스트 정보는 그림 1의 “Context Managing and Reasoning”에서 컨텍스트 관리를 위해 3단계의 처리 과정을 가진다: (1) Context Specification, (2) Context Operation, 그리고 (3) Context Service Recognition. 이러한 3단계 처

리 과정에서 컨텍스트 정보가 가공, 선별 과정을 거쳐 의미 있는 컨텍스트 인지 서비스를 제공한다. 이러한 유희원칙을 적용한 컨텍스트 정보 처리는 컨텍스트 정보에 대한 보다 직관적인 모델을 제공하고 다양한 도메인에서 사용되는 컨텍스트 정보에 대한 통합 기준을 제공하여 컨텍스트 정보 정의 및 관리에서 일관성 있는 통합과 활용을 가능하게 한다.



(그림 3) 컨텍스트 인지 정보 처리 프로세스

4. Case Study

본 논문에서는 제안 모델의 검증 및 분석을 위해 두 가지 시나리오를 활용한다. 다음의 두 가지 시나리오는 2장에서 기존의 모델들의 단점으로 제안한 이슈들을 모두 확인할 수 있는 시나리오이다. 즉, 제안 시나리오들은 특정 도메인에 의존하지 않으며, 많은 장치들과 상황들이 혼합되어 수집된 데이터들간의 상호운용성이 반드시 유지되어야 하며, 또한 컨텍스트의 인식과 함께 이를 통한 상황에 대한 반응이 필요한 시나리오이다. 즉, 컨텍스트의 응용과 조합이 적절히 조합되어야 하는 시나리오이다. 따라서 제안 시나리오들은 completeness 측면에서 볼 때 기존 모델들의 이슈들을 확인하고 본 논문에서 제안하는 모델을 검증하기에 적합한 시나리오이다.

4.1 시나리오 1

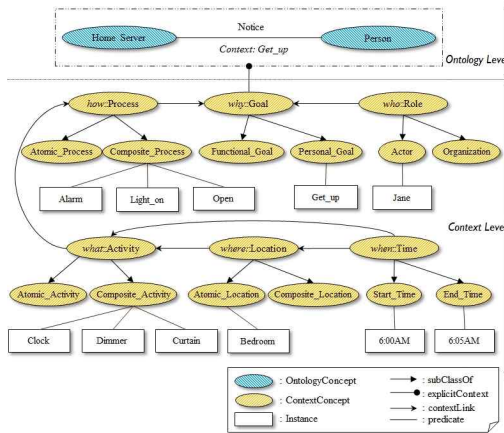
SYS Lab.에 근무하는 Jane은 내일 회사에서

아침9시에 회의가 있어 준비시간과 이동시간을 고려하여 아침 6시에 일어나야 한다. 이러한 Jane의 스케줄은 Home Server에서 관리하며 Jane이 아침 6시 일어날 수 있도록 Home Server는 시계에 알람을 맞춘다. 또한, 조명은 켜지도록 설정되며 커튼은 열리도록 디바이스를 설정한다. 따라서, 아침 6시가 되면 Home Server와 Jane사이에는 Get_up이라는 Context로 처리된다.

- **Ontology Level:** Home Server(온톨로지)와 Person(온톨로지)등이 존재한다. Home Server(온톨로지)는 홈에서 일어날 수 있는 여러 컨텍스트들이 미리 서버 내에 설정되어 온톨로지 인스턴스로 존재하게 된다. Person(온톨로지)는 가족구성원들에 대한 Profile 정보가 온톨로지로 구성된다. Home Server와 Person간의 관계는 Context 로 처리된다.

- **Context Level:** Home Server와 Person의 관계에서 발생한 Context를 유희원칙에 기반하여 처리한다. why::Goal은 Personal_Goal, Functional_Goal 등 하위 클래스를 가지며 최종적으로 처리되어야 할 Context를 말한다. 이 시나리오 상에서는 Jane을 깨우는 것이 why::Goal이 되며, Jane의 스케줄에 의해 발생한 Context이므로 Personal_Goal에 해당한다. 따라서 Personal_Goal은 Get_up을 인스턴스로 가진다. why::Goal은 행위의 대상으로써 who::Role과 관계를 가지며 who::Role은 하위클래스로 Actor와 Organization 등을 가진다. Actor에는 Get_up의 대상이 되는 Jane이 인스턴스로 존재한다. Get_up이라는 why::Goal을 실행하기 위하여 어떠한 작동이 필요한지 파악하기 위하여 how::Process와 디바이스 정보를 나타내는 what::Activity와 관계를 가진다. how::Process의 하위클래스에는 단독으로 프로세스를 실행하는 Atomic_Process와 복합적으로 프로세스를 실행하는 Composite_Process가 존재한다. what::Activity 또한 마찬가지로 하위클래스에 단독으로 동작하는 Atomic_Activity와 복합적으로 동작하는 Composite_Activity가 존재한다. Get_up의 경우 Activity들이 복합적으로 실행되어야하므로 Composite_Process는 Alarm, Light_on, Open을 인스턴스로 가진다. 또한, 시계, 조명, 커튼 등 여러 디바이스가 복

합적으로 동작해야 하므로, 시계, 조명, 커튼은 Composite_Activity의 인스턴스로 존재한다. 마지막으로, Context의 발생시점에 대한 정보를 위하여 장소에 대한 where::Location과 시간에 대한 when::Time와 관계를 가진다. where::Location의 하위 클래스에는 한 장소에 대한 정보를 처리하기 위한 Atomic_Location과 여러 장소에 대한 정보 처리를 위한 Composite_Location이 있다. Get_up이라는 Context는 Jane의 침실에서 단독으로 처리되므로 Bedroom은 Atomic_Location의 인스턴스로 존재한다. 알람이 울리는 시간인 when::Time의 하위클래스에는 Start_Time과 End_Time가 존재한다. 알람이 울리기 시작하는 시간인 Start_Time은 6:00AM을 인스턴스로 가지며, 알람이 끝나는 시간인 End_Time은 6:05AM을 인스턴스로 가진다.



(그림 4) 시나리오 1의 온톨로지 모델링

Example 1. Modeling based OntoCA

- OntoCA = {Concepts→Home_server, Person, Instances→Alarm, Light_on, Open, Clock, Dimmer, Curtain, Bedroom, Jane, SYS Lab, Contexts→Get_up}
- Context Model = {5W1H(Who, Why, How, What, Where, When,) }
- Who = {Role(Actor→Jane)}
- Why = {Goal(Personal_Goal→Get_up)}
- How = {Process(Composite_Process→Alarm, Light_on, Open)}

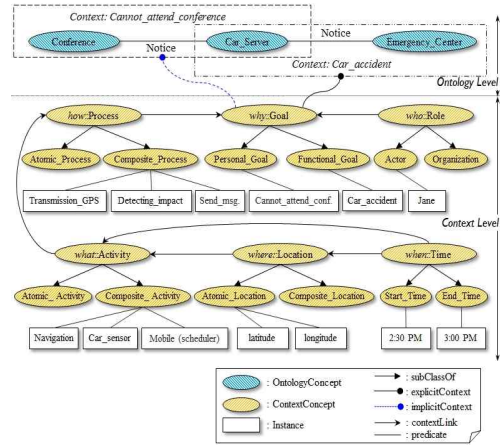
- What = {Activity(Composite_Activity→Clock, Dimmer, Curtain)}
- Where = {Location(Atomic_Location→Bedroom)}
- When = {Time(Start_Time→6:00AM, End_Time→6:05)}

4.2 시나리오 2

SYS. Lab.에서 근무하는 Jane은 오후 3시에 B도시에서 열리는 회의에 참석하기 위하여 차를 타고 고속도로를 달리다 사고가 났다. 차 센서를 통해 충격을 감지하고 네비게이션에서 GPS좌표를 측정하여 Car_Server는 사고지점에서 가장 가까운 응급센터(경찰서)에 전송하여 사고 발생을 알린다. 그리고 Jane의 스케줄러(모바일)가 컨택트를 추론하여 회의에 참석할 수 없다는 메시지를 회의 담당자에게 통보한다.

- **Ontology Level:** Car_server에는 차에 발생할 수 있는 여러 컨택트들에 대한 정보가 온톨로지 인스턴스로 구성되어 있다. Emergency_center는 사고와 관련한 여러 정보들이 온톨로지 인스턴스로 구성되어 있다. Car_server와 Emergency_center의 관계가 차 사고라는 Explicit Context로 처리되면 Jane이 기존에 수행중이던 회의참석에 대한 Context는 이들의 관계에 의해 영향을 받게 된다. 따라서, 기존에 존재하지 않았던 Car_server와 Conference(온톨로지) 사이에는 새로운 관계가 형성되며 회의불참이라는 context로 처리된다.
- **Context Level:** why::Goal은 Functional_Goal과 Personal_Goal을 하위클래스로 가진다. 시나리오 상의 Explicit Goal은 Car_server가 Emergency_center에 차 사고를 알리는 것이다. 뿐만 아니라, Jane이 참석하기 위해 이동중이었던 Conference에 대해서도 차 사고로 인한 불참을 알리는 것 또한 Implicit_Goal이 된다. 따라서 Functional_Goal은 차사고를 알리는 것이 되고, Personal_Goal은 회의불참이 된다. why::Goal은 행위의 대상으로 who::Role과 관계를 가진다. who::Role은 Actor와 Organization을 하위 클래스로 가지며, 시나리오상 Actor는 Jane을 인스턴스로 가진다. 또한 why::Goal은 디바이스 간에 어떠한 작동이 필요한지 파악하기 위하여 how::Process와

디바이스 정보를 나타내는 what::Activity와 관계를 가진다. how::Process의 하위클래스에는 단독으로 프로세스를 실행하는 Atomic_Process와 복합적으로 프로세스를 실행하는 Composite_Process가 존재한다. what::Activity 또한 마찬가지로 하위클래스에 단독으로 동작하는 Atomic_Activity와 복합적으로 동작하는 Composite_Activity가 존재한다. 차사고에 대한 Explicit_Goal을 처리하기 위하여 사고 위치 파악을 위한 Transmission_GPS와 사고유무 판단을 위한 Detecting_impact, 그리고 회의불참에 대한 Implicit_Goal을 처리하기 위하여 Send_message가 함께 Composite_Process로 처리되어야 한다. 또한, what::Activity의 작동 디바이스로는 Explicit_Goal 처리를 위한 Navigation, Car_sensor와 Implicit_Goal 처리를 위한 Mobile(Scheduler)가 Composite_Activity의 인스턴스로 존재한다. 마지막으로, Context의 발생시점에 대한 정보를 위하여 장소에 대한 where::Location과 시간에 대한 when::Time과 관계를 가진다. where::Location의 하위 클래스에는 한장소에 대한 정보를 처리하기 위한 Atomic_Location과 여러 장소에 대한 정보 처리를 위한 Composite_Location이 있다. 장소정보는 네비게이션이 측정한 위도와 경도를 Atomic_Location의 인스턴스로 가진다. 사고가 발생한 시간에 대한 정보는 when::Time가 처리한다. when::Time의 하위클래스에는 Start_Time과 End_Time가 존재하며, 사고발생 시간인 2:30PM은 Start_Time의 인스턴스가 되고, 3:00PM은 End_Time의 인스턴스가 된다.



(그림 5) 시나리오 2의 온톨로지 모델링

Example 2. Modeling based OntoCA

- OntoCA = {Concepts→Car_Server, Emergency_Center, Conference, Instances→Detecting_impact, Transmission_GPS, Send_message, Car_sensor, Navigation, Mobile_scheduler, Latitude, Longitude, Contexts→Cannot_attend_Conference, Car_accident}
- Context Model = {5W1H(What, Where, When, Why, Who, How) }
- Who = {Role(Actor→Jane)}
- Why = {Goal(Functional_Goal→Car_accident, Personal_Goal→Cannot_attend_Conference)}
- How = {Process(Composite_Process→Detecting_impact, Transmission_GPS, Send_message)}
- What = {Activity(Composite_Activity→Car_sensor, Navigation, Mobile_scheduler)}
- Where = {Location(Atomic_Location→Latitude, Longitude)}
- When = {Time(Start_Time→ 2:30PM, End_Time→3:00PM)}

5. 결론 및 향후 연구

본 논문에서는 기존 온톨로지 기반 컨텍스트 인지 모델들의 한계점을 보완하기 위하여 육차

원칙 온톨로지 기반의 컨텍스트 인지 모델을 제안하였다. 컨텍스트 인지 시스템의 전체적인 프레임워크와 함께 컨텍스트 인지 정보를 처리하기 위한 프로세스도 정의하였다. 또한 Case study를 통하여 컨텍스트 인지 모델 및 온톨로지를 정의하고 모델의 활용 가능성 및 검증을 수행하였다.

향후 연구로는 제안 모델의 구현을 통해 기존의 모델들과 정량적인 비교 평가를 수행함으로써 제안 모델과 기존 모델과의 차별성을 객관적으로 검증해야 할 것이다.

참 고 문 헌

- [1] G.D. Abowd, M. Ebling, G. Hung, H. Lei, and H-W. Gellersen "Context-aware computing," IEEE Pervasive Computing, Vol. 1, no. 3, pp. 22-23, 2002.
- [2] B. N. Schilit, N. L. Adams, and R. Want, "Context-aware computing applications," In IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US, 1994.
- [3] M. Samulowitz, F. Michahelles, and C. Linnhoff-popien, "Capeus: An architecture for context-aware selection and execution of services," In New developments in distributed applications and interoperable systems, Kluwer Academic Publishers, pp. 23 - 39, Krakow, Poland, September 17-19 2001.
- [4] T. Strang, "Service Interoperability in Ubiquitous Computing Environments," PhD thesis, Ludwig-Maximilians-University Munich, Oct. 2003.
- [5] W3C. Composite Capabilities / Preferences Profile (C/CPP), <http://www.w3.org/Mobile/CCPP>.
- [6] WAPFORUM. User Agent Profile (UAProf), <http://www.wapforum.org>.
- [7] A. Held, S. Buchholz, and A. Schill, "Modeling of context information for pervasive computing applications," In Proceedings of SCI 2002/ISAS 2002, 2002.
- [8] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," Computers and Graphics vol. 23, no. 6, pp. 893 - 901, 1999.
- [9] A. Schmidt, and K. V. Laerhoven, "How to Build Smart Appliances," IEEE Personal Communications, August 2001.
- [10] K. Cheverst, K. Mitchell, and N. Davies, "Design of an object model for a context sensitive tourist GUIDE," Computers and Graphics, vol. 23, no. 6, pp. 883-891, 1999.
- [11] C. Ghidini, and F. Giunchiglia, "Local models semantics, or contextual reasoning = locality + compatibility," Artificial Intelligence, vol. 127, no. 2, pp. 221 - 259, 2001.
- [12] F. Giunchiglia, "Contextual reasoning," Epistemologica - Special Issue on I Linguaggi e le Macchine, vol. 16, pp. 354-364, 1993.
- [13] J. McCarthy, "Notes on formalizing contexts," In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 555-560, San Mateo, California, 1993.
- [14] J. McCarthy, and C. Buva, "Formalizing context (expanded notes)," In Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language, Menlo Park, California, 1997.
- [15] CoBrA (Context Broker Architecture), <http://cobra.umbc.edu/>.
- [16] T. Gu, H.K. Pung and D.Q. Zhang, "A Middleware for Building Context-Aware Mobile Services," In Proceedings of IEEE Vehicular Technology Conference (VTC), 2004.
- [17] G. Biegel and V. Cahill, "A Framework for Developing Mobile, Context-aware Applications," IEEE International Conference on Pervasive Computing and Communications (PerCom), 2004.
- [18] D. Salber, Anind K. Dey and G. D. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," In Proceedings of CHI'99, Pittsburgh, PA, May 15-20, 1999.
- [19] T. G. GRUBER, "A translation approach to portable ontologies," Knowledge Acquisition, vol. 5, no. 2, pp. 199-220, 1993.
- [20] M. Uschold, and M. Gruninger, "Ontologies: Principles, methods, and applications," Knowledge Engineering Review, vol. 11, no. 2, pp. 93-155, 1996.
- [21] OWL (Web Ontology Language), <http://www.w3.org/TR/owl-features/>.
- [22] D. Zhang, T. Gu, X. Wang, "Enabling context-aware smart home with semantic technology," International Journal of Human-friendly Welfare Robotic Systems, vol. 6, no. 4, pp. 12-20, 2005.
- [23] DAML+OIL, <http://www.daml.org/>.

김진형



2006년 :고려대학교 대학원 (이학 석사)

2009년 :고려대학교 대학원 (이학 박사-데이터베이스)

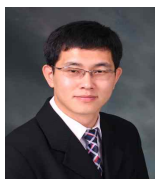
2008년~2009년 : 고려대학교 컴퓨터정보통신연구소

2009년~2011년 : LG 전자기술원 선임연구원

2011년~현재 : 한국과학기술정보연구원 연구원

관심분야 : 온톨로지, 시멘틱 웹, XML 데이터베이스

황명권



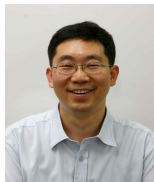
2006년 :조선대학교 대학원 (공학 석사)

2011년 :조선대학교 대학원 (공학 박사-데이터베이스)

2011년~현재 : 한국과학기술정보연구원 연구원

관심분야 : 온톨로지, 시멘틱 웹, 데이터베이스

정한민



1994년 :POSTECH 대학원 (공학 석사)

2003년 :POSTECH 대학원 (공학 박사-데이터베이스)

2004년~현재 : 한국과학기술정보연구원 책임연구원

2005년~현재 : 과학기술연합대학원대학교 겸임교수

2009년~현재 : 한국콘텐츠학회 이사

관심분야 : 시멘틱 웹, HCI, 자연어 처리