

## 다양한 도메인의 효율적 표현을 위한 온톨로지 기반의

### 다계층 컨텍스트 모델링

정민선<sup>0</sup>, 문마경, 김영봉, 염근혁  
부산대학교 컴퓨터 공학과

{msjung<sup>0</sup>, mkmoon, saram, yeom}@pusan.ac.kr

#### Multi-Layed Context Modeling Based on Ontology in an Effective Representation of Various Domains

Minsun Jung<sup>0</sup> Mikyong Moon Youngbong Kim Keunhyuk Yeom

Dept. of Computer Engineering, Pusan National University

#### 요 약

유비쿼터스 컴퓨팅 환경의 실현 가능성이 높아지면서 사용자가 존재하는 장소와 그곳의 환경에 따른 맞춤형 서비스의 제공이 요구된다. 이러한 맞춤형 서비스를 제공하기 위해서는 사용자 주변 환경을 인지 및 판단하여 서비스를 제공하는 소프트웨어가 필요하다. 소프트웨어가 환경을 인지하여 처리하려면 환경은 소프트웨어가 이해할 수 있도록 모델링 되고 언어로 표현되어야 한다. 기존의 context 모델링 방법은 특정 상황에 초점이 맞추어져 있어 다른 상황에서의 적용이 쉽지 않다. 본 논문에서는 다양한 도메인에 적용 가능한 다단계 상황 모델링 방법, 이를 ontology언어 OWL을 사용하여 나타내는 방법, 이것을 적용하기 위한 적응형 소프트웨어 개발 framework를 제시한다.

#### 1. 서 론

언제 어디서나 네트워크에 접속할 수 있는 유비쿼터스 컴퓨팅[1]을 위한 많은 기술들이 개발, 연구되고 있다. 사용자가 언제 어디서든 네트워크를 이용하여 환경을 변화시킬 수 있는 것은 물론 사용자에게 존재하는 장소와 동적인 주위 환경에 따라서 최적의 환경이 자동으로 구축되는 서비스가 제공되어야 한다. 다양하게 변화하는 외부 환경에 따라 적절한 서비스가 제공되기 위해서는 소프트웨어가 외부 환경 변화를 감지하고 이에 알맞은 기능을 수행해야 한다. 이를 위해 적응형 소프트웨어가 제안되었다. 적응형 소프트웨어란 외부 환경에 변화에 대한 자신의 행위를 평가하여 소프트웨어가 원래 의도한 것을 수행하지 못하거나, 더 나은 기능을 수행하는 것이 가능하다면 스스로 행위를 변경할 수 있는 소프트웨어를 말한다. 외부 환경에 영향을 받는 적응형 소프트웨어를 개발하기 위해서는 외부 상황 변화의 분석과 modeling이 필요하다. 상황 modeling에 대한 현재의 연구는 특정 상황만을 위한 modeling이 수행되고 있어 다른 상황 modeling에 적용하여 사용하기 어려우며 또한 상황을 modeling하는 것에만 초점이 맞추어져 있다. 본 논문에서는 다양한 상황에 적용가능하며 재사용이 쉬운 적응형 소프트웨어를 위한 2단계 상황 model과 model을 OWL을 이용하여 표현하는 것, 상황을 이용할 적응형 소프트

웨어를 위한 개발 framework을 제시한다.

#### 2. 관련 연구

##### 2.1 Modeling Context Information

적응형 소프트웨어가 동작하기 위해서는 외부 환경의 modeling이 필요하다. 물리적 또는 개념적인 외부 상황 개체들을 object 기반으로 modeling 하고 있다[2]. 특정 시나리오를 바탕으로 object를 추출하여 이들 간의 관계를 modeling하는 방법을 적용하고 있다. object간의 관계를 점진적으로 발전시키는 방법으로 복잡한 상황을 나타내고 있다. 이러한 방법은 하나의 환경에만 특화된 방법으로서 다양한 외부 환경에 적용하는데 어려움이 있다. 또한 외부 환경을 modeling하는 단계에서 연구가 끝난다. modeling된 결과를 실제 소프트웨어에 적용하기 위해서는 컴퓨터가 처리할 수 있는 형태로 modeling한 것을 표현하는 방법에 대한 더 나아가 연구가 필요하다. modeling된 상황에 대한 정보를 공유하기 위해 온톨로지를 기반으로하는 모델도 연구되고 있다[3]. Gaia system은 외부 환경을 first order logic을 이용하여 location을 중심으로 modeling을 하는 방법을 채택하고 있으나 소프트웨어가 관심 있는 상황 전체를 파악하기 힘들다[4]. 또한 modeling이 적용될 구조가 세분화되어 있지 않고 복잡한 상황을 처리하기에는 부족하다.

##### 2.2 Ontology

ontology는 인간과 프로그램이 특정 domain에 대한 지식

\* 본 연구는 한국과학재단

목적기초연구(R01-2003-000-10197-0)지원으로 수행되었음

을 공유하는데 도움을 주기 위해 사용된 개념화 명세서를 말한다[5]. ontology를 이용하여 상황을 modeling 하는 것은 몇 가지 장점을 갖는다. 첫째, 복잡한 domain에 대해 잘 정의된 ontology는 재사용이 가능하다. 둘째, low-level context로부터 high-level context로의 논리적 추론이 가능하다. W3C에 의해 제안된 Web Ontology Language(OWL)는 단지 사람에게 정보를 표시하는데 그치지 않고 정보의 내용을 직접 처리할 수 있는 어플리케이션을 구현하는데 활용될 수 있도록 설계된 언어이다 [6][7]. OWL은 표현력이 다른 OWL Lite, OWL DL, OWL Full으로 구성된다. 본 논문에서는 OWL Lite를 사용하여 context를 표현한다.

### 3. 다계층 Context Model

소프트웨어가 동작하는 환경은 그 종류가 다양하며 각각의 환경은 지속적으로 변화한다. 소프트웨어가 동적으로 변화하는 환경에 알맞은 동작을 수행하기 위해서는 외부 환경의 변화를 인식할 수 있어야 한다. 이를 위해 소프트웨어 외부 환경을 modeling하고 이것을 컴퓨터가 처리할 수 있는 형태로 표현하는 기법이 필요하다. 본 논문에서는 환경 modeling을 general level, specific level 2단계로 하는 방법을 제시하고 이를 OWL로 표현한다.

#### 3.1 General Level Context Model

소프트웨어 외부 환경을 context라 하며 context를 나타내는 요소들을 context entity라 한다. general level context model을 위해서는 소프트웨어의 적용이 필요한 여러 가지 상황(ex. smart home, intelligent lab, lecture room)을 가정하고, 공통적으로 나타나는 요소들을 추출한다. 필수 context entity는 person, activity, place, computing device 이다. context entity는 context를 표현하기 위한 data를 포함하며 이들 사이에는 여러 가지 관계가 존재한다. context는 하나 이상의 context entity들로 구성된다. 그림 1은 General Level Context Model을 보여준다.

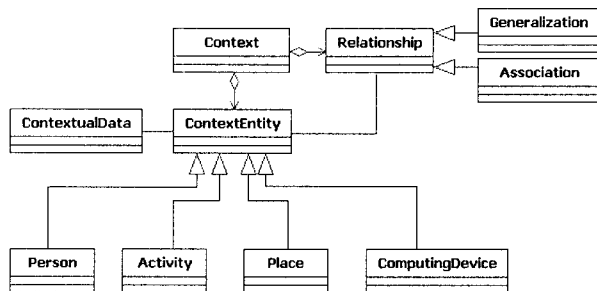


그림 1. General Level Context Model

#### 3.2 Specific Level Context Model

specific level context model은 general level context model을 특정 domain에 적용한 것을 말한다. 그림 2는 intelligent lab domain에 대한 specific level context model을 보여준다.

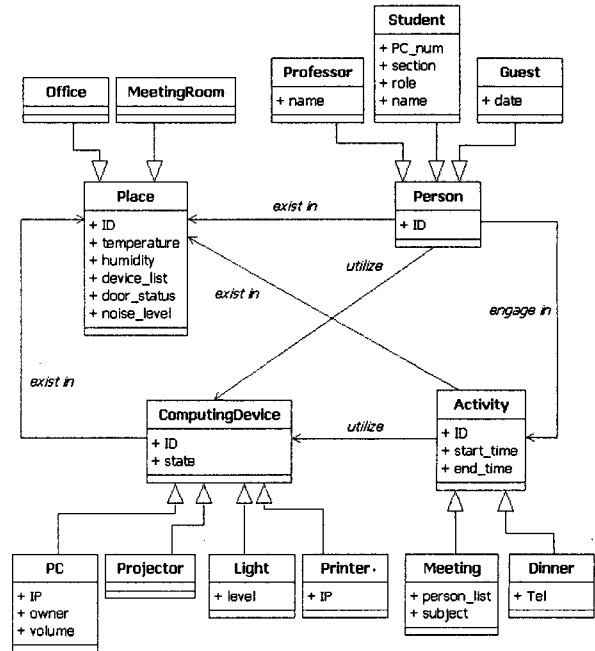


그림 2. Specific Level Context Model for Intelligent Lab

2단계 context model은 몇 가지 장점을 가진다. 첫째, 적응형 소프트웨어가 동작하기 위해 필요한 context를 빨리 이해할 수 있으며 한 단계 model 보다 접근하기 쉽다.

둘째, 특정 domain만을 위한 model이 아니므로 새로운 다양한 domain으로의 재사용이 가능하다.

#### 3.3 OWL로 표현

modeling된 context는 소프트웨어가 처리할 수 있는 형태로 변환되어야 한다. 컴퓨터가 data를 이해하고 자동으로 처리할 수 있도록 하기 위해 ontology가 제안되었다. 본 논문에서는 W3C에서 제안한 Web Ontology Language(OWL) Lite를 기반으로 modeling된 context를 나타낸다.

OWL은 class와 property를 표현하는 다양한 어휘들을 제공한다. context entity들은 class(owl:Class)로 표현되고, context entity들 간의 관계와 data들은 property로 표현된다. entity들 간의 계층은 owl:subClassOf를 이용해서 나타낸다. entity의 attribute들은 owl:DatatypeProperty, entity들 간의 관계는 owl:ObjectProperty를 사용하여 나타낸다. OWL의 built-in property를 이용하면 다

양한 상황을 표현하는 것이 가능하다. 그림 3은 Intelligent Lab domain에 대해 나타낸 OWL 표현의 일부이다.

```

<owl:Class rdf:ID="ContextEntity"/>
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="engageIn">
  <rdfs:domain rdf:resource="Person">
  <rdfs:range rdf:resource="Actiity">
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="utilize">
  <rdfs:domain rdf:resource="Person">
  <rdfs:range rdf:resource="ComputingDevice">
</owl:ObjectProperty> ...
<owl:DataProperty rdf:ID="ID">
  <rdfs:domain rdf:resource="Person">
  <rdfs:range rdf:resource="xsd:string">
</owl:DataProperty> ...
<owl:Class rdf:ID="Student">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <owl:differentFrom rdf:resource="Guest"/>
</owl:Class> ...
    
```

그림 3. Intelligent Lab을 위한 부분적 OWL 표현

4. 적응형 소프트웨어 개발 Framework

앞에서 소프트웨어가 동작하는 외부 환경을 modeling 하고 OWL을 이용하여 나타내었다. 이것만으로는 소프트웨어 개발의 어떤 부분에 모델링한 것이 적용되는지 알기 어렵다. 본 논문에서는 적응형 소프트웨어 개발을 위한 프레임워크를 설계하였다. 프레임워크는 Sensor or Reader, Context-aware Middleware(CaM), Context Event Framework(CEF), Self-Adaptive Application으로 구성된다. 그림 4는 적응형 소프트웨어 개발을 위한 Framework를 나타낸다.

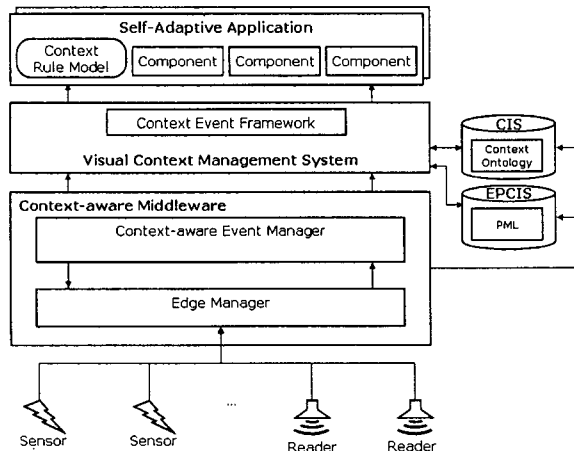


그림 4. 적응형 소프트웨어 개발을 위한 Framework

하위의 event는 상위로 올라가면서 그 형태가 바뀌게 된다. sensor 또는 reader로 부터의 event를 sensor event라 한다. sensor event는 middleware에 의해 수집되고 필터링 되어진다. middleware를 거치면 logical event로 변환되고 context event framework를 거치면 context event로 변환된다. 하위의 event가 여러 단계를 거치면 Self-Adaptive Application이 쉽게 이용할 수 있는 효율적인 형태로 변환된다. CaM은 edge들(sensor, reader)을 등록, 삭제, 관리하며 이들로부터 읽은 데이터를 수집하고 필터링하는 Edge Manager와 데이터를 바탕으로 context를 추론하는 Context-aware Event Manager로 구성된다. CEF는 적응형 소프트웨어 개발시에 event의 처리작업을 도와주는 library 형태를 말한다. Self-Adaptive Application은 추론된 context에 따른 Activity를 나타내는 Context Rule Model과 Component 들로 구성되어 있다.

5. 결론 및 향후 연구

본 논문에서는 유비쿼터스 컴퓨팅 환경에 필수적인 적응형 소프트웨어를 위한 context modeling 방법, OWL을 사용하여 이것을 표현하는 방법을 제시하고 적응형 소프트웨어 개발 framework을 설계하였다. context model을 general level, specific level 2단계로 수행함으로써 다양한 상황을 modeling하기가 쉬워졌다. 또한 context model을 OWL을 이용하여 표현하여 컴퓨터가 자동으로 처리할 수 있는 형태로 가공하였다. 적응형 소프트웨어 개발 framework을 통해 적응형 소프트웨어를 위해 필요한 전체 구조를 빠르게 파악할 수 있게 되었다. 향후 연구로는 ontology를 기반으로한 model을 framework에 실제로 적용한 system을 구성하는 것에 대한 연구가 필요하다.

6. 참고 문헌

- [1] M. Weiser, "The Computer for the 21st Century", IEEE Pervasive Computing 2002
- [2] Karen Henriksen, et al. "Modeling Context Information in Pervasive Computing Systems", Pervasive 2002.
- [3] COBRA-ONT, <http://cobra.umbc.edu/ontologies.html>
- [4] Anand Ranganathan, Roy H. Cambell, "An infrastructure form context-awareness based on first order logic", Personal and Ubiquitous Computing, Vol.7 Issue 6 pp.353-364, 2003
- [5] Gruber, T., "A Translation Approach to Portable Ontology Specifications", in Knowledge Acquisition Journal, Vol.5 pp.192-220, 1993
- [6] <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- [7] <http://www.w3.org/TR/owl-guide/>