

# 트리플 기반의 컨텍스트 모델

김은회\*, 최재영  
승실대학교 컴퓨터학부

e-mail: ehkim@ss.ssu.ac.kr, choi@ssu.ac.kr

## A Context Model using the Triples

Kim Eunhoe, Choi Jeayoung  
School of Computing, Soongsil University

### 요 약

유비쿼터스 컴퓨팅 환경은 공통적으로 재사용 가능하고, 확장성이 강력한 컨텍스트 모델이 필요하다. 본 논문에서는 엔티티(Entity), 컨텍스트 타입(Context Type), 값(Value)으로 구성된 트리플을 기반으로 한 유비쿼터스 컴퓨팅 환경의 컨텍스트를 모델링을 제안하였다. 컨텍스트 정보를 주어, 동사, 목적어에 해당하는 (Entity, Context Type, Value) 트리플로 표현함으로써 재사용성과 확장성을 제공한다.

### 1. 서론

유비쿼터스 컴퓨팅에 대한 연구가 본격적으로 활발히 이루어지기 이전인 1990년대 중반까지만 해도 컨텍스트는 자연어 처리와 HCI에서 주로 사용하던 개념이었다 [1]. 최근에는 모바일 컴퓨팅과 유비쿼터스 컴퓨팅의 발전과 더불어 사용자의 모바일 디바이스와 물리적인 주변 환경을 컨텍스트 정보로 인식하여 사용자에게 가장 적합한 서비스를 제공하기 위한 연구가 진행되고 있다. 이러한 컨텍스트 인식 응용프로그램에서는 공통적으로 재사용가능한 컨텍스트 모델이 필요하다 [2].

본 논문에서는 다양한 유비쿼터스 컴퓨팅 어플리케이션에서 공통적으로, 재사용 가능하며, 확장성이 쉬운 트리플 데이터 모델을 기반으로 한 컨텍스트 모델을 제안하고 실제 유비쿼터스 컴퓨팅 환경의 서비스 시나리오를 통하여 컨텍스트 모델의 유용성을 증명한다.

본 논문의 구성은 다음과 같다. 2장에서는 트리플 접근법의 대표적인 모델들과 장점을 논한다. 3장에

서는 유비쿼터스 컴퓨팅 환경에서 컨텍스트의 의미를 정의한다. 4장에서는 3장에서 정의한 컨텍스트를 트리플 접근법을 이용하여 모델링한다. 5장에서는 유비쿼터스 어플리케이션 시나리오에 컨텍스트 모델을 실제 적용하여 컨텍스트를 표현한다. 6장에서는 관련연구와 트리플 기반 컨텍스트 모델을 비교 설명한다. 7장에서는 결론을 내린다.

### 2. 트리플(Triple, 3-tuples) 접근법

트리플 기반의 데이터 모델의 대표적인 모델로는 RDF (Resource Description Framework) [3], First Order Predicate [2] 등이 있다. RDF과 First Order Predicate 모두 세 개의 정보쌍인 트리플로 정의되는데, 이러한 정보들은 일반 문장의 주어, 서술어, 목적어에 대응된다. RDF 데이터 모델에서 트리플의 관계를 그래프로 표현하면 그림 1과 같다. 즉 RDF의 statement는 (property, resource, object)의 트리플로 표현되며 property는 서술어, resource는 주어, object는 목적어에 각각 대응한다.



그림 1. RDF statement의 트리플 그래프

RDF는 W3C의 표준으로 시맨틱 웹의 자원에 대한 메타 정보를 표현하기 위한 가장 기초적인 프레임워크이다. RDF는 웹에서 뿐만 아니라, 응용 분야가 매우 다양하다. 자원의 개념을 식별할 수 있는 사물로 일반화하여 웹에 존재하지 않는 자원들에 대한 메타데이터를 기술할 수 있기 때문이다.

First Order Predicate는 지식을 형식적으로 표현하기 위한 한 방법으로써 강력한 표현력을 제공한다. Predicate는 임의의 object의 특성과 관계를 기술할 수 있으며, 오늘날의 모든 지식은 First Order Predicate로 기술할 수 있는 것으로 알려져 있으며, 추론과 같은 인공지능적인 처리에 유용하다.

### 3. 유비쿼터스 컴퓨팅에서 컨텍스트

모바일 컴퓨팅에서는 사용자가 이동 컴퓨팅 장비를 사용하여 로케이션에 관계없이 서비스를 제공하거나 제공받을 수 있는 컴퓨팅 기술에 초점이 맞춘다 [4]. 따라서 모바일 컴퓨팅에서 가장 중요한 정보는 사람들과 사물들의 location 정보이다 [5]. 따라서 컨텍스트는 사용자 가까이에서 있는 사람들과 객체들의 집합, 그리고 시간에 따른 이러한 객체들의 변화를 가리킨다.

유비쿼터스 컴퓨팅 환경은 실세계의 도처에 산재하는 객체들에 극소형의(눈에 보이지 않을 만큼 작은) 컴퓨터를 심어 지능화하고, 이러한 컴퓨터들을 무선으로 서로 연결하여, 언제 어디서나 자연스럽게 편리하게 이용하도록 함으로써 컴퓨터의 사용을 증진시키는 컴퓨팅 환경을 말한다 [6]. 따라서 유비쿼터스 컴퓨팅 환경에서 컨텍스트는 사람이나 사물의 위치, 아이덴티티, 시간, 온도, 소음 등을 참조할 때 사용한다 [1].

본 논문에서 컨텍스트는 유비쿼터스 컴퓨팅 환경에서 엔티티의 상태 또는 상황을 설명하는 정보이고, 엔티티를 센싱하거나 또는 엔티티와 관련된 정보에서 유도되며, 이러한 엔티티는 사용자, 어플리케이션을 포함한 실세계의 모든 무형, 유형의 객체로 정의한다.

이론을 포함한 실세계의 모든 무형, 유형의 객체로 정의한다.

### 4. 트리플 기반의 컨텍스트 모델

본 논문에서는 (Entity, Context Type, Value)의 세가지 정보로 이루어진 트리플을 사용한 컨텍스트 모델을 제시한다 (그림 2). 트리플 각각의 요소는 순서대로 (subject, predicate, object)에 각각 대응한다.

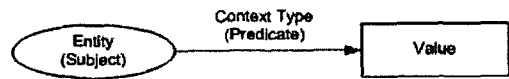


그림 2. 트리플 기반 컨텍스트 모델의 그래프 표현

Entity는 컨텍스트 정의에 따라 사용자, 어플리케이션을 포함한 실세계의 모든 무형, 유형의 객체이다. 즉, 사람, 어플리케이션, 스케줄, TV, 전화, 냉장고, 컴퓨터, PDA, 프린터, 창문, 빌딩이나 집, 장소 등이 모두 엔티티가 될 수 있다. Entity는 컨텍스트 정보의 주체에 해당하므로 subject에 대응된다.

Context Type은 Entity의 상황과 상태의 종류를 설명한다. 예를 들면 사람의 위치 정보를 기술하는 컨텍스트 정보에서 Context Type은 위치이다. location 외에 많이 사용하는 컨텍스트 타입으로 Activity, Status, Nearby, Identity, Temperature, Sound, Weather 등이다. Context Type은 Entity의 속성을 기술하므로 Predicate에 대응된다.

Value는 실제로 엔티티의 상황과 상태를 설명하는 값이다. 예를 들면, 사람의 Activity 컨텍스트의 값으로는 meeting, eating, sleeping, watchTV 등이 있다. TV의 Status 컨텍스트의 값으로는 on, off 등이 있다. 어떤 장소의 온도 컨텍스트에 대한 값으로는 30°C와 같은 값이 Value가 될 수 있다. 이와 같이 컨텍스트 정보의 Value는 실제 기본적인 데이터를 가질 수도 있지만 엔티티도 값이 될 수도 있다. 예를 들면, (A, location, livingRoom)이라는 컨텍스트 정보는 livingRoom이라는 엔티티를 Value로 가지게 된다.

컨텍스트 정보는 (Entity, ContextType, Value) 트리플로 표현할 수 있지만, 어플리케이션에서 좀더 의미있는 컨텍스트 정보를 표현할 필요가 있다. 예

를 들면 “A가 침실에서 자고 있다”라는 상황을 컨텍스트 정보로 표현하기 위해서는 (A, Activity, Sleeping), (A, LocatedIn, Bedroom) 두개의 트리플을 연결할 필요가 있다. 이를 위해서 AND, OR, NOT과 같은 기본적인 논리 연산자를 이용한다. 따라서 위의 컨텍스트 정보는 (A, Activity, Sleeping) AND (A, LocatedIn, Bedroom)으로 표현한다.

5장에서는 트리플 기반의 컨텍스트 모델을 실제 유비쿼터스 컴퓨팅 환경의 서비스 시나리오에 적용하며, 트리플 기반의 컨텍스트 모델의 재사용성과 확장성 등을 논한다.

### 5. 시나리오

사람들의 일상은 어떤 목적을 위해서 실행되는 일이 많고 이러한 일들의 대부분은 일정한 작업 흐름을 가지고 있으며, 매일같이 반복된다. 다음은 유비쿼터스 홈네트워크상에서 회사원 A씨가 기상할 때, 실행되는 일련의 서비스들을 시나리오로 구성하였다.

시나리오: 늦은 시간에 잠자리에 드는 A씨는 이른 아침 기상을 위하여 홈네트워크 상에서 다음과 같은 일련의 서비스를 실행하는 컨텍스트어웨어 어플리케이션 시스템(Context-aware Application System: CAS)의 도움을 받는다. CAS는 A의 스케줄에 따라 6시가 되면 기상 도우미 어플리케이션을 실행시킨다. 그리고 제일 먼저 ‘A씨가 잠을 자고 있는지 깨어 있는지’의 상황을 파악한다. 깨어 있다면 알람 서비스를 생략하고 다음 순번의 서비스로 이동한다. A씨가 잠을 자고 있다면 Alarm 서비스를 실행시킨다. Alarm 서비스는 A씨가 잠을 깬 때까지 계속 실행된다. 따라서 Alarm 서비스는 루프를 계속 실행하고 A가 잠에서 깨었다는 이벤트가 발생했을 때 루프를 종료한다. A씨가 잠을 깨었다는 것을 인식한 CAS는 Alarm 서비스 루프를 빠져나와서 다음 단계의 일들을 진행한다. 다음에 진행할 서비스는 블라인드를 올리는 서비스를 필두로 홈네트워크의 정보가전 및 제어 네트워크의 기구를 제어하는 서비스들과 Information Display 서비스로 크게 나누워진다. 위의 서비스들은 서로 의존 관계가 없으며 사용자에게 동시성적인 서비스를 지원하기 위하여 병렬로 실행시킨다. 먼저 Blind up 서비스를 실행시킨다. 그리

고 날씨를 검색하여 비가 오거나 온도가 너무 낮다면 공기청정 서비스를 실행시키고, 그렇지 않다면 창문을 열기위한 window open 서비스를 실행시켜서 신선한 공기를 유입시킨다. 그리고 실내의 밝기를 센싱하여 어둡다면 light on 서비스를 실행시킨다. 한편 Information Display 서비스는 A씨의 현재 위치에서 가장 가까운 display 장치에 A씨의 일과중 중요 스케줄을 출력한다. 이런 일련의 모든 서비스를 실행한후 CAS는 기상 도우미 어플리케이션을 종료한다.

시나리오는 그림 3과 같이 다이어그램으로 표현될 수 있으며, 각 서비스들의 시작 조건과 서비스 선택 등을 표현할 때, 본 논문에서 제시한 트리플 기반의 컨텍스트 모델을 사용하여 컨텍스트 정보를 표현하였다.

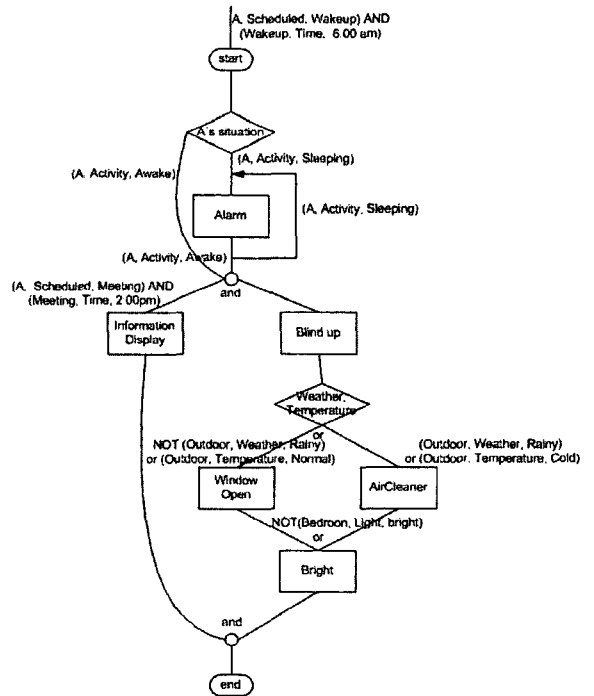


그림 3. 기상 도우미 시나리오 다이어그램

시나리오에 필요한 컨텍스트 정보를 (Entity, Context Type, Value)의 트리플로 표현하면 표 1과 같다. 트리플 기반의 컨텍스트 모델은 표 1과 같이 다양한 컨텍스트의 표현이 가능하다. 또한 정의한 트리플을 다른 도메인에서 재사용할 수 있으며, AND, OR, NOT과 같은 논리 연산자를 통하여 복잡

한 컨텍스트로 쉽게 표현할 수 있다.

표 1. 트리플 기반의 컨텍스트 정보 표현

Context Type	Entity	Value	Examples
Activity	Person	Sleeping Awake	(A, Activity, Sleeping) (A, Activity, Awake)
LocatedIn	Person	Bedroom Livingroom Bathroom Kitchen Outdoor	(A, LocatedIn, Bedroom) (A, LocatedIn, Outdoor)
BlindStatus	Blind	Down Up	(Blind, BlindStatus, Down)
Window-Status	Window	Opened Closed	(Window, WindowStatus, Opened)
Weather	Location	Cloudy Rainy Sunny Snowy	(Seoul, Weather, Sunny)
Temperature	Location	Cold Normal Hot	(Bedroom, Temperature, Hot)
Light	Location	Bright Normal Dark	(Livingroom, Light, Dark)
Schedule	Person	Activity	(A, schedule, meeting)
Time	Location, Activity	hh:mm am/pm	(Seoul, Time, 06:00 am) (meeting, Time, 02:00 pm)

## 6. 관련 연구와의 비교

유비쿼터스 컴퓨팅 환경의 컨텍스트에 대한 많은 모델들이 연구되어 왔다. 컨텍스트 모델은 컨텍스트의 사용 도메인과 시스템의 요구사항에 따라 또는 모델링 관점에 따라 매우 다양하다.

컨텍스트 개념에 대한 이해를 돕기 위한 E-R 다이어그램에 기반한 컨텍스트 모델[7]이 있고, 객체지향적인 방법론에 따라 컨텍스트의 속성과 분류를 기준으로 컨텍스트를 모델링하는 방법[8]이 있다. 그리고 이기종 웹 환경에서 시스템간 컨텍스트 정보의 상호 운영성을 지원하기 위한 온톨로지 기반의 컨텍스트 모델, 컨텍스트 정보를 인공지능적으로 처리하기 쉽도록 지식 관점의 컨텍스트 모델링을 수행하기도 한다 [2].

본 논문에서 제시한 트리플 기반의 컨텍스트 모델은 유비쿼터스 어플리케이션에서 공통적으로 컨텍스트를 표현하기 위한 모델이며, 확장성이 강력한 모

델이다. 왜냐하면 모든 지식의 표현은 주어, 동사, 목적어로 표현이 가능하기 때문이다. 또한 시멘틱 웹이 온톨로지 언어들이 대부분 트리플을 사용하는 RDF를 기반으로 한 언어들이기 때문에 온톨로지로 표현하기 용이하여 시스템간 컨텍스트 정보의 공유가 용이하다.

## 7. 결론

유비쿼터스 컴퓨팅 환경에서 공통적으로 재사용 가능하며 확장성이 가능한 컨텍스트 모델이 필요하다. 본 논문에서는 유비쿼터스 컴퓨팅 환경의 트리플 기반의 컨텍스트 모델을 제안하였다. 컨텍스트 정보를 지식을 표현하는 가장 기본적인 주어, 동사, 목적어에 해당하는 (Entity, Context Type, Value) 트리플로 표현함으로써 재사용성과 확장성을 지원한다. 또한 시스템간 지식공유를 위한 온톨로지로 구축이 용이한 모델을 제공한다. 향후 트리플 기반의 컨텍스트 모델을 기반으로 유비쿼터스 컴퓨팅 시스템간 컨텍스트 정보 공유를 위한 온톨로지 컨텍스트 모델을 개발할 예정이다.

## 참고문헌

- [1] Anind k. Dey: Understanding and Using Context, Personal and Ubiquitous Computing, Vol 5, Issue 1 (2001).
- [2] Anand Ranganathan, Roy H. Campbell, An Infrastructure for context-awareness based on first order logic, Personal and Ubiquitous Computing, Volume 7, Issue 6 (2003).
- [3] W3C: RDF/XML Syntax Specification, W3C Recommendation, Feb 2004.
- [4] Guanling Chen, David Kotz, A Survey of Context-Aware Mobile Computing Research, Technical Report, TR200381, Dartmouth College (2000).
- [5] Albrecht Schmidt, et. al. Advanced interaction in context. In Proceedings of First International Symposium on Handheld and Ubiquitous Computing, Germany, Sept. 1999.
- [6] M. Weiser, "The Computer for the 21st Century," IEEE Pervasive Computing, Jan.-Mar. 2002.
- [7] Karen Henriksen, Jadwiga Indulska, Andry Rakotonirainy: Modeling Context Information in Pervasive Computing Systems, Pervasive 2002, LNCS 2412 (2002) 167-180.
- [8] Kappel Gerti, et al.: "Customisation for Ubiquitous Web Applications - A Comparison of Approaches", International Journal of Web Engineering and Technology (IJWET), Inderscience Publishers, viewg, January 2003.