

## 워크플로우를 위한 컨텍스트 모델링

김은희, 한주현, 최재영  
송실대학교 컴퓨터학부

{ehkim<sup>o</sup>, jhhan}@ss.ssu.ac.kr, choi@comp.ssu.ac.kr

### A Context Model for the Workflow in Ubiquitous Computing

Kim Eunhoe, Han Juhyun, Choi Jeayoung  
School of Computing, Soongsil University

#### 요 약

유비쿼터스 컴퓨팅 환경의 워크플로우에서 컨텍스트 정보를 표현하기 위한 요구사항을 분석하고 이러한 요구사항을 만족하기 위한 트리플 구조(주어 동사 목적어) 기반의 단순 컨텍스트 모델과 합성 컨텍스트 모델을 제안한다. 제안한 워크플로우를 위한 컨텍스트 모델은 표현력이 강력하고, 확장이 쉽고, 재사용이 가능하며 인공지능적인 처리에 용이한 정보 구조를 제공한다.

#### 1. 서 론

유비쿼터스 컴퓨팅에서 서비스들은 서로 연계된 흐름을 가지며, 컨텍스트의 변화에 따라 사용자의 개입 없이 자동으로 적절한 서비스가 선택되어 실행되거나, 병렬적 혹은 반복적으로 실행되어야 한다. 유비쿼터스 컴퓨팅에서 서비스의 자동화를 구축하기 위하여 전통적인 컴퓨팅 환경의 서비스 자동화에 적용하던 워크플로우 모델을 적용하려는 시도가 이루어지고 있다[1].

본 논문에서는 유비쿼터스 컴퓨팅 환경의 다양한 워크플로우 프로세스에서 컨텍스트 정보를 쉽게 표현할 수 있도록 하기 위하여 확장성 있고, 표현력이 강력한 컨텍스트 모델을 제시한다.

본 논문의 구성은 다음과 같다. 2절에서는 컨텍스트의 의미와 범주를 정의한다. 3절에서는 유비쿼터스 컴퓨팅 환경의 워크플로우를 위한 컨텍스트 모델의 요구사항을 분석하고, 이러한 요구사항을 만족하는 단순 컨텍스트와 합성 컨텍스트 모델을 제시한다. 5절에서는 관련 연구와 본 논문에서 제시한 컨텍스트 모델을 비교 설명하며 6절에서 결론을 맺는다.

#### 2. 컨텍스트 정의

Schilit와 Theimer가 “컨텍스트 인지(Context-aware) 컴퓨팅”을 정의하면서 소개된 컨텍스트는 “위치, 근접해 있는 사람들과 객체들의 집합, 그리고 시간에 따른 이러한 객체들의 변화”를 가리켰다[2]. 이러한 정의는 컨텍스트를 실체를 들어 정의하기 때문에, 리스트에 없는 어떤 정보가 컨텍스트 정보인지 판단하는 명확한 개념을 제공하지 못하므로 좀 더 오퍼레이션 관점의 컨텍스트 정의가 필요하다[3]. 본 논문에서는 다음과 같이 컨텍스트를 정의한다.

“컨텍스트는 엔티티의 상황을 기술하는 정보로서, 엔티티를 센싱하거나, 다른 컨텍스트 정보로부터 유도 될 수 있다. 엔티티는 유비쿼터스 컴퓨팅 환경의 사용자와 어플리케이션을 포함

한 사용자와 어플리케이션 사이의 상호작용과 관련 있다고 고려되는 사람, 장소, 객체들이다.”

#### 3. 컨텍스트 모델

##### 3.1. 워크플로우를 위한 컨텍스트 모델의 요구사항

워크플로우란 전체적인 또는 부분적인 비즈니스 프로세스의 자동화를 말한다. 비즈니스 프로세스에 참여하는 사람 또는 어플리케이션 사이에서는 어떤 액션을 위해서 문서, 정보, 태스크들이 정해진 절차에 따라 전달된다[4].

워크플로우는 프로세스 정의 단계에서 프로세스를 구성하는 액티비티들과 액티비티들의 처리 절차 및 시작 및 종료 조건 등과 같은 정보를 표현한다. 따라서 프로세스 정의시 컨텍스트 정보를 표현하고, 워크플로우 관리시스템에서 컨텍스트 정보를 유지 관리 할 수 있어야 한다. 따라서 유비쿼터스 컴퓨팅 환경의 워크플로우를 위한 컨텍스트 모델의 요구사항은 다음과 같다.

첫째, 워크플로우 시스템은 많은 프로세스들을 가지므로 각 프로세스별로 컨텍스트 정보를 유지 관리해야 한다.

둘째, 유비쿼터스 컴퓨팅의 워크플로우 컨텍스트 모델은 전이 조건, 이벤트, 제약 조건을 표현할 수 있어야하므로 표현력이 풍부해야 한다.

유비쿼터스 컴퓨팅 환경의 워크플로우에서 컨텍스트 정보가 영향을 미치는 범주는 크게 경로 결정과 상태 전이이다. 워크플로우의 전이란 한 액티비티가 완전히 끝나고 제어권을 다른 액티비티에 넘기는 시점에 발생한다. 전이는 조건없이 한 액티비티가 끝나면 자동으로 다른 액티비티에 제어권을 넘기는 경우도 있지만 조건에 의해서 다음의 액티비티가 실행되기도 한다. 또한 여러 가지 조건에 따라 실행해야 할 액티비티들을 동적으로 결정하기도 한다. 이러한 경로 결정과 상태 전이에 영

항을 미치는 요소는 전이 조건(pre-condition, post-condition), 이벤트, 각종 제약 조건 등이다. 따라서 컨텍스트 모델은 이러한 요소들을 표현할 수 있도록 강력한 표현력을 제공해야 한다.

셋째, 컨텍스트 정보는 인공지능적인 처리를 통하여 고수준의 컨텍스트 정보를 추론할 수 있는 정보 구조를 가져야 한다. 컨텍스트 정보는 환경을 구성하는 엔티티에 대한 단순한 사실을 표현하는 저수준의 컨텍스트로부터, 이러한 저수준의 컨텍스트를 조합하고 인공지능적인 처리 과정을 거친 사용자의 의도와 같은 고수준의 컨텍스트까지 그 수준이 매우 다양하다. 따라서 인공지능 적인 처리가 용이한 정보구조를 가져야 한다.

넷째, 유비쿼터스 컴퓨팅 환경은 도메인에 따라 다양한 컨텍스트 정보를 표현해야 하므로 확장성이 있어야 한다. 일반적으로 많이 사용하는 컨텍스트 종류는 location, Activity, Nearby, Status, Temperature, Time 등이 있다. 워크플로우 프로세스들은 목적에 따라 다양한 액티비티들을 가질 수 있다. 이들과 관련있는 컨텍스트 정보 또한 다양하기 때문에 새로운 컨텍스트 정보를 일관성 있게 표현할 수 있는 컨텍스트 모델을 가져야 한다.

다섯째, 컨텍스트 모델은 다양한 도메인에서 재사용될 수 있어야 한다. 지금까지의 컨텍스트 모델은 특정 어플리케이션에 종속적인 컨텍스트 모델이 대부분이므로 어플리케이션이 달라지면 컨텍스트 모델 또한 바뀌어야 한다는 단점이 있다. 따라서 컨텍스트 모델은 워크플로우의 다양한 도메인에서 재사용할 수 있어야 한다.

따라서 본 논문에서는 컨텍스트 히스토리, Triple 구조의 단순 컨텍스트, 합성 컨텍스트를 모델링하고 이러한 컨텍스트 모델이 유비쿼터스 환경의 컨텍스트 모델의 요구사항을 어떻게 만족하는지를 설명한다. 본 논문에서는 논문에서 제시하는 컨텍스트 모델에 대한 개념을 돕기 위하여 개념적 차원의 클래스 다이어그램을 사용한다.

### 3.2. 컨텍스트 히스토리 모델

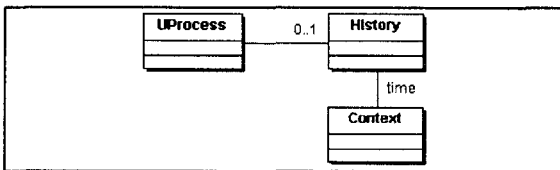


그림 1. 컨텍스트 히스토리 모델

유비쿼터스 컴퓨팅의 워크플로우 시스템은 해당 도메인에 따라 다양한 프로세스들을 가지며 컨텍스트 정보는 각 프로세스마다 따로 구분되어 관리 되어야 한다. 따라서 그림 1과 같이 컨텍스트는 각 프로세스당 하나의 컨텍스트 히스토리를 가지며 시간을 기준으로 컨텍스트 정보를 가진다. 따라서 컨텍스트 히스토리를 사용하여 각 프로세스별로 컨텍스트 정보를 유지 관리하게 된다.

컨텍스트 정보는 시간을 기준으로 과거, 현재, 미래의 컨텍스트 정보로 구분할 수 있다. 과거 컨텍스트 정보는 추론이나 확

습, 데이터 마이닝 같은 처리를 통해 고수준의 컨텍스트를 유추하기 위하여 사용된다. 미래 컨텍스트 정보는 워크플로우의 프로세스를 예측하고, 각종 자원을 예약하기 위해 사용된다. 현재 컨텍스트 정보는 워크플로우의 상태 전이에 가장 직접적인 영향을 미치는 요소이다.

### 3.3. Triple 구조의 단순 컨텍스트 (Primitive Context) 모델

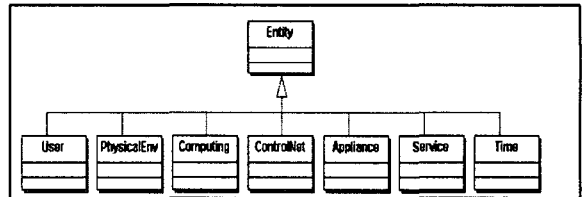


그림 2. 엔티티

유비쿼터스 환경의 워크플로우에서 사용하는 컨텍스트의 타입은 워크플로우 도메인에 따라서 그 종류가 서로 다르며 매우 다양하다. 컨텍스트의 정의에 따라 결국 컨텍스트 정보란 엔티티에 대한 상황 정보이므로 컨텍스트 정보의 주체는 엔티티이다.

유비쿼터스 환경에서 모든 객체는 모두 엔티티가 될 수 있고, 엔티티의 종류는 워크플로우의 도메인에 따라 좌우된다. 예를 들면 사람, 어플리케이션, 사물, 디바이스, 네트워크, 장소, 시스템 시계 등, 인간의 삶의 모든 환경을 구성하는 유형 또는 무형의 객체들이 엔티티가 될 수 있다. 그림 2는 워크플로우의 다양한 엔티티들을 표현한다.

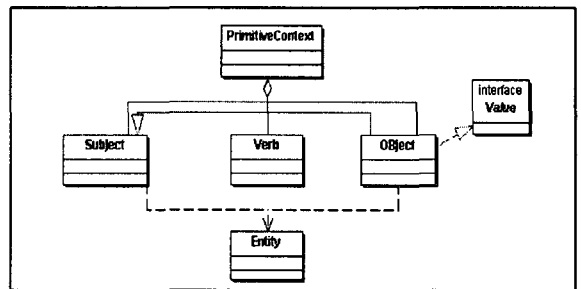


그림 3. 단순 컨텍스트(PrimitiveContext) 모델

컨텍스트 정보를 표현하기 위하여 본 논문에서는 ‘(주어, 동사, 목적어)’의 Triple 구조를 사용하여 컨텍스트 정보를 표현한다. 컨텍스트 정보의 주체가 되는 엔티티 정보는 주어로, 컨텍스트 타입은 동사로, 실제 값은 목적어로 표현한다. 본 논문에서는 (주어, 동사, 목적어)의 트리플로 표현하는 컨텍스트 정보를 단순 컨텍스트라고 부른다. 그림 3은 단순 컨텍스트와 주어 동사 목적어와 엔티티의 연관관계에 대한 모델이다.

이러한 (주어, 동사 목적어)의 Triple구조는 지식을 표현하는 가장 단순한 표현 방법으로 First Order Predicate 그리고 온톨로지 언어의 DAML, OWL 등의 기본 프레임워크인

RDF(Resource Description Framework)[5]에서 정보를 Triple 구조로 표현한다. 왜냐하면 (주어, 동사, 목적어)로 모든 자원 또는 사실들을 기술할 수 있기 때문이다[5]. 따라서 (주어, 동사, 목적어) Triple 구조의 단순 컨텍스트는 다양한 도메인의 컨텍스트 정보 쉽게 표현할 수 있으며, 확장성이 좋은 표현법을 제공한다.

그리고 (주어, 동사, 목적어)의 트리플 구조는 지식을 표현하는 일반적인 방법이므로 다양한 도메인에서 재사용될 수 있다. 또한 컨텍스트 정보를 (주어, 동사, 목적어)의 Triple 구조로 표현하면 추론과 같은 인공지능적인 처리의 대부분이 (주어, 동사, 목적어)의 데이터 표현을 기본으로 하므로 별도의 데이터의 변환이 필요없게 된다. 따라서 저수준의 컨텍스트 정보로부터 고수준의 컨텍스트 정보를 유추하기 위한 인공지능 적인 처리에 유용한 정보구조를 지원하게 된다.

예를 들면 방의 온도가 낮으면 난방서비스 액티버티를 실행시키는 서브프로세스가 있다고 할 때, '방의 온도가 낮다.' 컨텍스트 정보를 난방서비스 액티버티의 pre-condition에 명시해야 한다. Temperature 컨텍스트 정보를 표현할 때, 단순 컨텍스트 모델에 따라 (room301, Temperature, low)로 표현한다. room301은 장소를 나타내는 엔티티이며, 컨텍스트의 주체에 해당하므로 주어로, 온도는 컨텍스트 타입에 해당하므로 동사로, low는 실제 엔티티의 상황 값에 해당하므로 목적어로 표현한다.

A의 위치가 집밖일 때 보안서비스 액티버티를 실행시키는 서브 프로세스가 있다고 할 때, 'A의 위치가 집밖이다'라는 컨텍스트 정보를 보안서비스의 pre-codition에 해당하는 이벤트 정보로 명시해야 한다. 따라서 이를 표현 할 때, 단순 컨텍스트 모델에 따라 (A, location, outdoor)로 표현한다. A는 컨텍스트의 주체인 엔티티이므로 주어로, location은 컨텍스트 타입을 가리키므로 동사로 표현하고, outdoor는 실제 엔티티의 상황을 설명하는 값에 해당하므로 목적어로 표현한다. 지금의 예처럼 목적어에 해당하는 실제 엔티티의 상황을 기술하는 값은 상황에 따라 다른 장소를 나타내는 엔티티인 outdoor처럼 엔티티가 될 수도 있다.

3.4. 합성 컨텍스트 (Composite Context) 모델

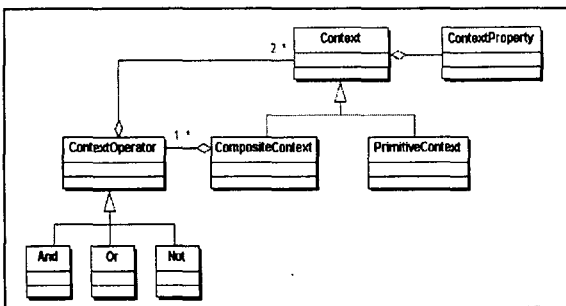


그림 4. 합성 컨텍스트와 단순 컨텍스트의 관계 모델

워크플로우의 전이 조건, 이벤트, 각종 제약조건에 컨텍스트 정보를 표현하기 위해서는 좀 더 복잡한 컨텍스트 정보를 표현할 수 있는 방법이 필요하다. 그림 4와 같이 단순 컨텍스트를

AND, OR, NOT과 같은 연산자를 이용하여 복잡한 컨텍스트를 표현하도록 한다. 단순 컨텍스트가 AND, OR, NOT으로 연결된 컨텍스트를 합성 컨텍스트라고 한다. 예를들면 "A가 침실에서 자고 있다."면 작업 분기를 하는 프로세스가 있다면, 이러한 컨텍스트 정보는 동일한 엔티티 즉 A라는 엔티티에 대하여 '침실에 있다'와 '자고 있다'와 같은 두가지의 상황을 표현해야 하기 때문에 단순 컨텍스트로 표현이 불가능하다. 따라서 합성 컨텍스트를 이용하여, (A, location, bedroom) and (A, activity, sleeping)와 같이 표현한다. AND, OR, NOT과 같은 연산자를 이용한 컨텍스트 정보의 표현은 복잡한 조건식과 제약조건, 이벤트를 쉽게 표기할 수 있는 강력한 표기법을 제공한다.

4. 관련연구

컨텍스트 모델에 대한 많은 연구가 진행되고 있다. 컨텍스트 모델은 컨텍스트들간의 관계를 개념적으로 쉽게 이해할 수 있도록 도와주는 모델[6]부터 실제로 어플리케이션에서 컨텍스트를 표현하고 처리하기 위한 컨텍스트 모델[7]까지 매우 다양하다. 본 논문과 가장 유사한 컨텍스트 모델은 GAIA의 First-Order-Logic(FOL)기반의 컨텍스트 모델[7]이다. 하지만 FOL기반의 컨텍스트 모델은 ContextType(S, V, O) 같은 표현법을 사용하므로 표현력은 좀 더 강력하지만 실제로 4가지의 요소를 사용한 컨텍스트 모델이기 때문에 정보공유를 위한 은톨로지 기반의 모델과는 호환이 어렵다는 단점이 있다. 본 논문은 FOL 기반의 모델보다는 표현력은 약하지만 컨텍스트 타입을 동사로 표현하는 트리플구조를 가지므로 은톨로지의 변환이 쉬우며, 인공지능적인 처리에 용이하다는 장점이 있다.

5. 결론

본 논문에는 유비쿼터스 컴퓨팅 환경의 워크플로우를 위한 컨텍스트 모델을 제시하였다. 제시한 컨텍스트 모델은 (주어, 동사, 목적어)의 Triple구조의 단순 컨텍스트 모델과 연산자를 연결된 합성 컨텍스트 모델, 그리고 컨텍스트 히어로 모델을 제안하였다. 제안한 컨텍스트 모델은 워크플로우 요구사항을 수용하며 확장성 있고, 재사용성이 가능하다. 향후 워크플로우간의 정보 공유를 위하여 Triple 구조 기반의 컨텍스트 모델을 기반으로 은톨로지 기반의 컨텍스트 모델을 개발할 예정이다.

6. 참고 문헌

1. S.-Y. Hwang et al., "Personal Workflow Management in Support of Pervasive Computing," Proc. of the 2nd Int'l. Conf. On Mobile Data Management, Hong Kong.(2001).
2. Guanling Chen, David Kotz, A Survey of Context-Aware Mobile Computing Research, Technical Report, TR200381, Dartmouth College (2000).
3. Anind k. Dey: Understanding and Using Context, Personal and Ubiquitous Computing, Vol 5, Issue 1 (2001).
4. WfMC, Workflow Management Coalition Terminology & Glossary, WfMC-TC-1011, (1999).
5. W3C: RDF/XML Syntax Specification, W3C Recommendation, Feb 2004.
6. Karen Henriksen, Jadwiga Indulska, Andry Rakotonirainy: Modeling Context Information in Pervasive Computing Systems, Pervasive 2002, LNCS 2412 (2002) 167-180.
7. Anand Ranganathan, Roy H. Campbell, An Infrastructure for context-awareness based on first order logic, Personal and Ubiquitous Computing, Volume 7, Issue 6 (2003).