한국정보전자통신기술학회논문지(jkiiect)'17-10, Vol.10 No.5
논문 17-10-05-380 http://dx.doi.org/10.17661/jkiiect.2017.10.5.380

IoT 환경에서 온톨로지 기반의 상황정보 모델

김은회*, 서유화**

A Situation Information Model based on Ontology in IoT Environment

Eunhoe Kim*, Yuhwa Suh**

요 약 IoT 환경의 서비스는 상황정보를 끊임없이 수집하여, 상황에 적합한 행동을 수행해야 한다. 따라서 수집한 상황정보를 표현할 수 있는 방법이 필요하다. 본 논문에서는 IoT 환경의 상황정보를 표현할 수 있도록 온톨로지 기반의 상황정보 모델을 제안한다. 제안하는 모델은 온톨로지 기반이므로 의미적인 상호운용성을 지원한다. 또한 다양한 IoT 도메인의 상황정보를 모델링하기 위하여 상위 온톨로지를 작성한다. IoT 환경을 구성하는 사람, 환경, 사물을 클래스로 표현하고, 그 상황을 나타내는 속성을 정의하여 일관성 있게 상황정보를 기술하므로 이해하고 사용하기 쉽다. 또한 상황정보는 시시각각 변하는 동적인 상황을 반영해야 하므로 상황정보 온톨로지의 유효성을 판단할 수 있도록 상황정보의 생성시간 및 생존시간을 모델링하여 제공하는 특성을 가진다. 제안하는 온톨로지 모델은 OWL을 사용하여 기술하며, 구축한 온톨로지를 기반으로 서비스를 기술할 수 있다.

Abstract The services of the IoT environment should constantly collect situation information, and perform appropriate actions according to the situation. Therefore, there is a need for a method that can express collected situation information. In this paper, we propose a situation information model based on ontology for IoT environment. Since the proposed model is ontology based, it supports semantic interoperability. We also build an upper-level ontology to model common situation information of various IoT domains. It is easy to understand and use because it expresses situation information consistently by expressing person, environment, and thing constituting IoT environment as class and defining properties indicating the situation. In addition, since the situation information need to reflect dynamic situation, it has a feature to model the creation time and the life time of the situation information so as to judge the validity of the information. The proposed ontology model is described using OWL, and the service can be described based on the constructed ontology.

Key Words : Context, IoT Environment, Ontology, Sematic Interoperability, Situation Information

1. 서론

IoT(Internet of Things) 기술은 환경을 구성하는 사람뿐만 아니라 사물들을 인터넷에 연결하고, 상황정보를 수집하여 사용자에게 적합한 서비스를 지원하는 기술이다. IoT 환경의 서비스를 제공하기 위해서는 사용자 상황 정보뿐만 아니라 주변 사물과 환경의 상황정보를 끊임없이 수집하여 서비스를 수행할 조건이 만족하는지를 검사해야 한다[1]. 따라서 수집되는 IoT 환경의 상황정보들은 표현할 수 있는 방법이 필요하다.

본 논문에서는 IoT 환경의 상황정보를 표현할 수 있도록, 온톨로지 기반의 상황정보 모델을 제안한다. 온톨로지(Ontology)는 '공유된 개념에 대한 정형적이고 명확한 명세' 라고 정의된다[2]. 온톨로

*Department of Software Engineering, Seoil University
**Corresponding Author : Department of Information and Communication Engineering, Seoil University (5syh@seoil.ac.kr)
Received September 13, 2017 Revised September 24, 2017 Accepted October 12, 2017

지는 정형적이면서, 정보의 의미를 풍부하게 표현할 수 있고, 논리적인 특성을 기술할 수 있는 방법을 제공한다. 또한 사람뿐만 아니라 기계도 이해할 수 있는 의미적인 상호운용성(Interoperability)을 제공하는 기술이다. 따라서 IoT 환경의 사람 및 환경, 다양한 사물들로부터 수집되는 상황정보의 의미를 논리적으로 표현하기 적합하며, 사람뿐만 아니라 IoT를 구성하는 다양한 기계들 사이에서 상황정보에 대한 의미적인 상호운용성을 지원할 수 있다.

본 논문에서 제안하는 온톨로지 기반의 상황정보 모델은 대표적인 온톨로지 언어인 OWL[3] 를 사용하여 기술하며, 다양한 IoT 도메인의 상황정보를 모델링하기 위하여 상위 온톨로지를 작성한다. IoT 환경을 구성하는 사람, 환경, 사물을 클래스로 표현하고, 그 상황을 나타내는 속성을 정의하여 일관성 있게 상황정보를 기술하므로 이해하고 사용하기 쉽다. 또한 온톨로지의 유효성을 판단할 수 있도록 상황정보의 생성시간 및 생존시간을 모델링하여 제공하는 특성을 가진다.

본 논문의 2장에서는 IoT 환경에서 온톨로지 기반의 상황정보 모델의 관련 연구를 분석한다. 3장에서는 다양한 도메인에서 공통적으로 적용될 수 있는 상위 온톨로지를 정의한 후, 스마트 홈 도메인에서 서비스 시나리오를 위한 도메인 온톨로지를 구축한다. 4장에서는 구축한 온톨로지를 검증하기 위하여 서비스들의 조건을 SPARQL로 검색하고, Jena 룰을 사용하여 서비스를 기술한다. 5장에서는 결론을 맺는다.

2. 관련 연구

IoT 환경의 온톨로지 기반의 상황정보 모델로 GOMs(Generic Ontology Models)[4], SISO(Semantic-based IoT Service Ontology)[5], MOnCa 온톨로지[6] 등이 제안되었다. GOMs, SISO, MOnCa 온톨로지는 모두 여러 도메인이서 공통적으로 사용되는 요소들을 상위 온톨로지로 구현하고 상위 온톨로지의 하위 온톨로지로 도메인 온톨로지를 구축하여 다양한 도메인의 서비스를 지원하는 상황 정보 모델을 제공한다.

GOMs는 서비스에 필요한 범용적인 요소들을 독립적인 버티컬 온톨로지로 정의한 후, 서비스의 요구사항이 바뀌거나 새로운 서비스를 정의할 때 범용으로 정의한 온톨로지의 하위에 필요한 온톨로지를 추가하는 방식으로 여러 도메인의 다양한 서비스에서 공유할 수 있도록 한다. 또한 각각의 버티컬한 온톨로지들을 속성과 관계를 통해서 서로 연결하여 서비스를 정의하는 클로스 버티컬 온톨로지 모델을 정의한다[4].

SISO는 상황 정보 표현을 위해서 상황을 구성하는 요소를 해당 상위 클래스의 하위 클래스로 구성하는 온톨로지를 구축한다. 또한 서비스를 위한 상황 정보 조건들을 온톨로지의 클래스로 정의하고, 클래스 추론을 통하여 상태 정보를 적합한 클래스로 추론한다[5].

GOMs와 SISO는 IoT 서비스 모델에 중점을 두었고, 서비스를 위한 IoT 상황정보를 함께 표현하여 서비스에 필요한 조건을 추론한다. 따라서 상황정보를 표현할 때, 사람, 사물과 환경 클래스에 대한 속성으로 상황정보를 표현하기보다, 사람, 사물과 환경 클래스의 하위 클래스를 사용하여 필요한 상황정보를 정의함으로써 하위 온톨로지로 갈수록 클래스 분류 체계가 매우 복잡하여 이해하기 어렵다.

MOnCa 온톨로지는 스마트 폰에서 지능형 서비스 제공을 목적으로 하는 베이스 온톨로지, 코어 온톨로지, 도메인 온톨로지, 콘텍스트 온톨로지, 링크된 오픈 데이터를 사용한다[6]. 서비스를 지원하기 위한 상황 정보 모델링에 초점을 두었으며 컨텍스트 온톨로지를 별도로 정의하여 온톨로지 추론을 통해 고수준의 컨텍스트를 유추하여 사용하는 특징이 있다. MOnCa의 베이스 온톨로지는 상황정보를 표현하기 위해 사람, 사물과 환경을 클래스로 표현하고 그 상황을 속성으로 표현하므로 온톨로지가 이해하기 쉽다. 그러나 동적인 IoT 상황정보의 유효성을 판단할 수 있는 정보는 제공하지 않는다.

본 논문에서 제안하는 IoT 환경의 상황정보 모델은 GOMs, SISO, MOnCa처럼 여러 IoT 도메인의 상황정보를 모델링하기 위하여 도메인간 공통적으로 사용되는 요소들에 대하여 상위 온톨로지를 구축하고, 도메인에 특정된 상황정보는 상위 온톨로지의 하위에 정의하여 온톨로지의 재사용성을 높인다. 또한 MOnCa처럼 상황정보를 표현하기 위하여 사람, 환경, 사물을 클래스로 정의하고 상황을 나타내는 속성을 정의하여 상황정보 모델을 이해하고 사용하기 쉽다. IoT 서비스는 정의한 온톨로지를 기반으로 Jena 룰을 사용하여 기술한다, Jena rule reasoner를 사용하여 고수준의 상황정보를 추론하며, 복잡한 상황정보를 서비스 조건으로 기술하여 서비스를 기술 한다. 상황정보의 유효성을 판단할 수 있도록 상황정보의 생성시간 및 생존시간을 메타데이타로 표현하는 특징이 있다.

3. IoT 환경의 상황정보 모델

3.1 상위 온톨로지

IoT 환경은 홈, 오피스, 병원, 공항, 도시 등 여러 크고 작은 도메인에 적용되며, 다양한 서비스들을 제공하고 있다. 따라서 IoT 상황정보 온톨로지 모델은 여러 도메인에 적용될 수 있도록 작성되어야 한다. 온톨로지의 재사용성을 높이기 위한 온톨로지 작성 방법으로 상위 온톨로지(Upper level ontology)를 구축하여 사용하는 방안이 있다. 상위 온톨로지는 여러 도메인에 적용될 수 있는 일반적인 카테고리들을 정의한 온톨로지이다[7]. 따라서 다양한 IoT 도메인에서 공통적으로 사용될 상위 온톨로지를 정의하고, 상위 온톨로지를 기반으로 IoT 도메인에 특정된 온톨로지를 구축하면 새로운 도메인의 온톨로지를 쉽게 작성할 수 있고, 다른 IoT 도메인의 온톨로지들과의 통합도 쉽다는 장점이 있다.

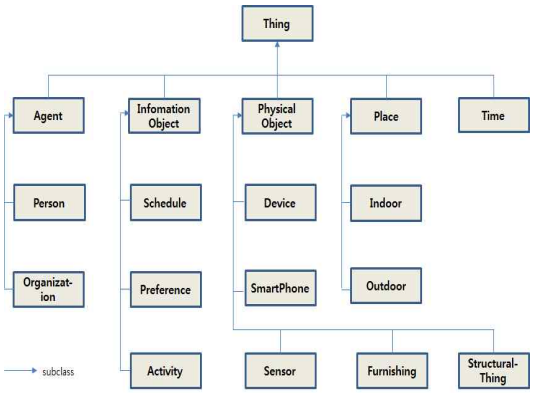


그림 1. IoT 환경을 지원하기 위한 상위 온톨로지
Fig. 1. Upper Level Ontology for IoT Environment

본 논문에서는 IoT의 다양한 도메인에서 일반적으로 공통적으로 사용하는 Agent, InformationObject, PhysicalObject, Time, Place 카테고리들을 그림 1과 같이 상위 온톨로지로 작성한다. 에이전트의 하위 클래스로는 Person, Organization가 있다. InformationObject의 하위 클래스는 Preference, Schedule, Activity 가 있다. PhyscialObject의 하위 클래스는 Device, SmartPhone, Sensor, Furnishing, StructuralThing으로 구성된다. Place의 하위클래스는 Indoor, Outdoor로 구성된다. 작성된 상위 온톨로지를 기반으로 3.2절에서는 홈 도메인에서 IoT 서비스를 정의한 후, 상위 온톨로지의 하위 온톨로지로 홈 도메인에서 상황 정보 모델을 구축한다.

3.2 홈 도메인에서 상황정보 온톨로지

스마트 홈 환경은 IoT 기술이 적용되는 가장 주목 받는 분야 중의 하나다[8]. 사용자의 편의, 방범과 보안, 에너지 관리, 각종 가전 기구 모니터링 및 제어 등을 위해 IoT 기술이 적용된다. 본 논문에서는 그림 2와 같이 스마트 원룸을 대상으로 사용자의 기상 편의를 제공하는 스마트 홈 기상 알람 서비스 시나리오와 사용자가 오랜 시간 외출을 하였을 때, 빈 집이 범죄의 표적이 되지 않도록 집을 보호하기 위한 스마트 홈 방범 서비스 시나리오를 정의한다.

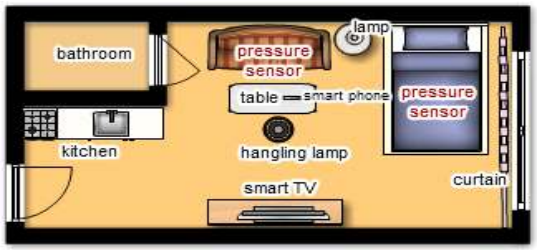


그림 2. 시나리오 환경: 스마트 원룸
Fig. 2. The environ. of the Scenarios: Smart One-room

스마트 홈 기상 알람 서비스: 사용자는 먼저 기상 시간을 오전 5시로 설정한다. 스마트 홈 기상 알람 서비스는 사용자가 설정한 시간에 맞추어 스마트 홈의 상황 정보를 점검한다. 먼저 사용자가 침실에 있는지를 체크한다. 그리고 침대에 압력센서의 값을 검사하여 침대가 사용 중인지를 체크하고, 침대 램프가 켜져 있는지를 확인한다. 위의 조건이 다 만족하면 사용자가 침실에서 취침중이라고 판단하고, 핸드폰의 알람을 울리고, TV를 켜고, 커튼을 오픈하고, 중앙 조명을 켜는 서비스를 가동한다.

스마트 홈 방범 서비스: 사용자의 스케쥴에서 출장을 확인한다. 이번에 사용자는 일주일간 출장을 간다. 집주인이 장기간 집을 비운다는 것을 외부에 표시나지 않게 하기 위해서 스마트 홈 방법 서비스는 저녁 7시가 되면 컨튼을 내리고 불을 켠다. 밤 11시가 되면 소등을 한다. 아침 7시가 되면 커튼을 올린다. 방범 서비스를 사용자가 돌아오는 날까지 지속한다.

그림 3은 스마트 홈 도메인에서 정의한 서비스 시나리오를 중심으로 상황정보 모델을 표현한다. 3.1절에서 정의한 상위 온톨로지를 기반으로 상위 온톨로지의 하위에 서비스에 필요한 상황정보를 그림 3과 같이 정의한다.

스마트 홈의 사용자는 kim 이라는 인스턴스로 표현된다. 현재 시간은 current라는 Time 클래스의 인스턴스로 표현되며, date, time, season이라는 속성을 갖는다. kim의 위치를 나타내는 상황정보는 Person 클래스의 kim 인스턴스와 Indoor 클래스의 인스턴스 사이에 locatedIn 속성으로 표현한다. 온톨로지 정보를 RDF 트리플로 표현하면, (kim, locatedIn, bedroom) 으로 표현할 수 있다. PhysicalObject 클래스는 Indoor 클래스와 locatedIn 관계를 가진다. 따라서 bedroom에 있는 침대, 베드 램프 등의 상황정보는 (bed, locatedIn, bedroom), (bed_lamp, locatedIn, bedroom)으로 표현된다.

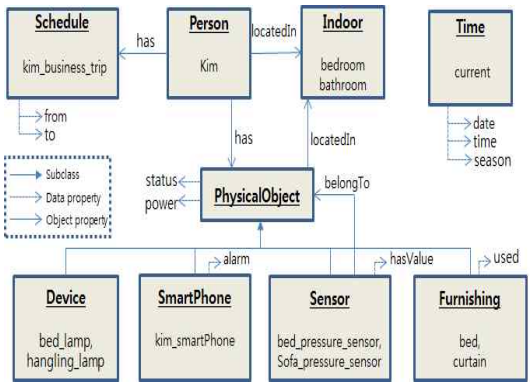


그림 3. 스마트 홈 온톨로지
Fig. 3. An Ontology for the Smart Home

센서는 보통 물리적인 객체에 소속되어 존재하므로, Sensor 클래스는 PhysicalObject 클래스와 belongTo 관계를 가진다. '침대 압력 센서가 침대에 소속되어 있다' 는 상황정보를 표현하기 위해서는 (bed_pressure_sensor, belongTo, bed) 라고 기술한다. PhysicalObject는 status, power와 같은 속성을 갖는다. 따라서 '베드 램프의 불이 켜있다.' 와 같은 상황정보는 (bed_lamp, power, "on") 과 같이 표현한다. 가구를 나타내는 Furnishing 클래스는 used라는 속성을 가진다. Furnishing 클래스의 인스턴스인 침대에 대해 '침대가 사용중이다' 라는 상황정보는 used 속성을 사용하여, (bed, used, "true") 로 기술한다. 커튼의 상태 정보는 status 속성을 사용하여 (curtain, status, "down") 과 같이 표현한다.

Schedule 클래스는 스케쥴을 표현하기 위한 클래스 온톨로지이다. 스케쥴의 기간을 표현하기 위하여 from, to 속성을 가진다. Schedule 클래스의 인스턴

스인 kim_business_trip이 kim이라는 사용자의 출장 스케쥴을 위해 정의되었을 때, '출장 시작일은 2017년 7월 24일 9시 이다.', '출장 종료일은 2017년 7월 28일 19시이다' 와 같은 상황 정보는 (kim_schedule_trip, from, "2017-07-24T09:00:00"), (kim_schedule_trip, to, "2017-07-28T19:00:00") 로 표현한다. from, to 속성은 온톨로지의 data property로 표현되며 치역(range)는 XML Schema의 datetime 자료형을 갖는다.

3.3 IoT 상황정보의 메타 정보

IoT에서의 서비스는 상황정보를 기반으로 하여 서비스의 조건을 검사하고 적절한 서비스를 제공하게 된다. 상황정보는 IoT 환경의 시시각각 변하는 동적인 상황을 반영해야 하므로 상황정보가 유효한지를 판단할 수 있어야 한다[9]. 이러한 상황정보의 유효성을 표현하는 정보들은 IoT 서비스의 조건을 명시할 수 있는 상황정보라기 보다 상황정보를 부가적으로 설명하는 데이터이므로, 이러한 정보들을 상황정보의 메타데이타라고 정의한다[9]. 상황정보 메타데이타는 상황정보를 생성할 때 같이 생성하게 되어 상황정보 수집기에 의해 사용된다. 본 논문에서는 이러한 상황정보의 메타 데이터를 표현하기 위해서 인스턴스에 주석을 추가할 때 사용하는 owl annotation 속성을 사용한다.

상황정보의 유효성을 판단하기 위해서는 해당 상황정보가 언제 생성되었는지, 그리고 언제 까지 유효한지에 대한 정보를 기술해야 한다. 이를 기술하기 위하여 그림 4, 그림 5와 같이 productionTime, lifetime annotation 속성을 정의한다. kim_business_trip 스케쥴이 2017년 7월 17일 9시에 생성되었다면, (kim_business_trip, productionTime, "2017-07-17T09:00:00") 로 표현된다. kim_business_trip 스케쥴은 생성된 후, 그 정보가 1시간(3600초)까지 유효하며, 1시간 뒤에는 스케쥴 리소스로부터 다시 상황정보 값을 모니터링 해야 한다면, (kim_business_trip, lifetime, 3600) 이라고 메타데이타를 표현한다.

```
<owl:AnnotationProperty rdf:about="http://www.semanticweb.org/seoil/
ontologies/2017/7/IotOnt#productionTime">
    <rdfs:domain
    rdf:resource="http://www.semanticweb.org/seoil/ontologies/2017/7/I
otOnt#InformationObject"/>
    <rdfs:domain
    rdf:resource="http://www.semanticweb.org/seoil/ontologies/2017/7/I
otOnt#PhysicalObject"/>
</owl:AnnotationProperty>
```

그림 4. 'productionTime' annotation 속성
Fig. 4. 'productionTime' annotation property

```
<owl:AnnotationProperty rdf:about="http://www.semanticweb.org/seoil/
ontologies/2017/7/IotOnt#lifeTime">
    <rdfs:domain
     rdf:resource="http://www.semanticweb.org/seoil/ontologies/2017/
7/IotOnt#InformationObject"/>
    <rdfs:domain
     rdf:resource="http://www.semanticweb.org/seoil/ontologies/2017/
7/IotOnt#PhysicalObject"/>
</owl:AnnotationProperty>
```

그림 5. 'lifeTime' annotation 속성
Fig. 5. 'lifeTime' annotation property

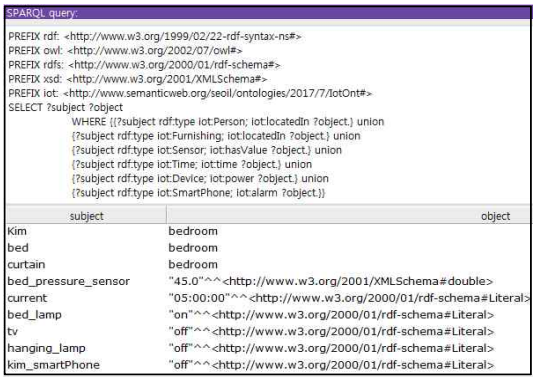
4. 구현 결과

3장에서 제안한 IoT 환경의 스마트 홈 상황정보 모델을 구현하기 위해서 OWL을 사용한다. OWL은 W3C의 워킹 그룹에서 만든 권고안이며, 현재 온톨로지를 구축하기 위한 언어로 많이 사용된다. RDF, RDFS, DAML+OIL 등과 같은 다른 온톨로지 언어들과 비교할 때 의미와 로직에 대한 표현력이 뛰어나다[2]. 온톨로지 구축에는 OWL 편집기인 protege 5.2.0[10]을 사용한다. 스마트 홈 기상알람 서비스와 방범 서비스는 스마트 홈 상황정보 온톨로지를 사용하여 Apache Jena[11]의 룰(rule)로 기술한다. Apache Jena는 온톨로지 API와 추론 API를 제공하는 시멘틱 웹 프레임워크로서 OWL을 잘 지원할 뿐 아니라 OWL reasoner를 제공하고 있어, OWL 온톨로지 구축 및 추론에 많이 사용되며, 온톨로지 추론을 위해서 제공하는 Jena 룰은 온톨로지를 사용하여 서비스를 기술하는데 적합하다[11]. 구현한 온톨로지에서 상황정보를 검색하기 위하여 온톨로지 질의 언어인 STARQL[12]을 사용한다. STARQL은 SQL과 유사한 문법을 가지고 있으므로 쉽게 배울 수 있고 온톨로지 질의에 적합한 언어이다.

본장에서는 3장에서 제안한 IoT 환경의 스마트

홈 상황정보 모델을 구현한 OWL 온톨로지가 스마트 홈 서비스에 필요한 상황정보를 잘 기술할 수 있는지 검증하기 위하여 온톨리지를 구축한 후, 서비스에 필요한 상황정보를 STARQL로 검색하여 그 결과를 제시한다. 또한 상황정보 온톨로지를 사용하여 실제로 서비스를 기술할 수 있는지를 검증하기 위하여 구현한 온톨로지를 사용하여 Apache Jena 룰로 서비스를 기술하고 그 결과를 제시한다. Apache Jena 룰로 기술한 서비스는 Jena 추론기에 의해 서비스 조건이 만족할 때, 서비스 액션과 연관된 해당 온톨로지를 오류없이 추론할 수 있어야 한다. 이를 검증하기 위하여 RuleDerivation 프로그램을 작성하여 그 실행결과를 제시한다. RuleDerivation 프로그램은 온톨로지를 데이터 모델로 입력받고, Jena Rule로 기술된 서비스 룰을 입력받아 Jena RuleReasoner에 의해 서비스 조건이 만족되었을 때, 룰 기반 추론을 하고 수행한 추론의 유추과정을 보여주는 프로그램이다. 상황정보 메타데이타 온톨로지는 서비스를 기술하는데 사용되기보다 상황정보를 수집하여 제공하는 시스템에서 상황정보를 수집할 때 활용하는 정보이다. 따라서 상황정보 메타데이타 온톨로지를 검증하기 위해서는 STARQL로 검색하여 해당 상황정보 메타데이타 온톨로지가 검색 가능해야 한다. 이를 위하여 실제 XML 인코딩으로 상황정보 메타데이타가 기술한 예를 들고, STARQL로 검색한 결과를 제시한다.

그림 6은 스마트 홈 기상 알람 서비스 시나리오를 구현한 온톨로지에서 관련 주요 상황정보를 온톨로지 질의 언어인 SPARQL을 사용하여 검색한 결과이다. Kim, bed, curtain 인스턴스의 위치가 bedroom 이고, bed_pressure_sensor의 값이 45.0, 현재 시간은 5시, bed_lamp는 on 상태이고, tv와 hanging_lamp의 power는 off 상태이고, kim_smartPhone의 알람은 off 상태임을 나타내고 있으며, 3.2절에서 정의한 스마트 홈 기상 알람 서비스의 상황정보를 잘 표현한다는 것을 나타내고 있다.



SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX iot: <http://www.semanticweb.org/seoil/ontologies/2017/7/IotOnt#>
SELECT ?subject ?object
        WHERE {{?subject rdf:type iot:Person; iot:locatedIn ?object.} union
        {?subject rdf:type iot:Furnishing; iot:locatedIn ?object.} union
        {?subject rdf:type iot:Sensor; iot:hasValue ?object.} union
        {?subject rdf:type iot:Time; iot:time ?object.} union
        {?subject rdf:type iot:Device; iot:power ?object.} union
        {?subject rdf:type iot:SmartPhone; iot:alarm ?object.}}
```

subject	object
Kim	bedroom
bed	bedroom
curtain	bedroom
bed_pressure_sensor	"45.0"^^<http://www.w3.org/2001/XMLSchema#double>
current	"05:00:00"^^<http://www.w3.org/2000/01/rdf-schema#Literal>
bed_lamp	"on"^^<http://www.w3.org/2000/01/rdf-schema#Literal>
tv	"off"^^<http://www.w3.org/2000/01/rdf-schema#Literal>
hanging_lamp	"off"^^<http://www.w3.org/2000/01/rdf-schema#Literal>
kim_smartPhone	"off"^^<http://www.w3.org/2000/01/rdf-schema#Literal>

그림 6. 기상 알람 서비스 상황정보 온톨로지 검색
Fig. 6. Situation information ontology retrieval for getting-up alarm service

```
@prefix  iot: <http://www.semanticweb.org/seoil/ontologies/2017/7/IotOnt#>.

[bed_used: (iot:bed_pressure_sensor iot:belongTo ?a),
           (iot:bed_pressure_sensor iot:hasValue ?b),
           greaterThan(?b, 20.0)
           -> ( ?a iot:used "true"^^rdfs:Literal )]

[smart_alarm: (iot:kim iot:locatedIn iot:bedroom),
              (iot:bed iot:used "true"^^rdfs:Literal),
              (iot:bed iot:locatedIn iot:bedroom),
              (iot:bed_lamp iot:power "on"^^rdfs:Literal),
              (iot:current iot:time "05:00:00"^^rdfs:Literal)
                -> (iot:tv power "on"^^rdfs:Literal ),
              (iot:kim_smartPhone iot:alarm "on"^^rdfs:Literal),
              (iot:curtain iot:status  "up"^^rdfs:Literal),
              (iot:hanging_lamp iot:power "on"^^rdfs:Literal )]
```

그림 7. 기상 알람 서비스 룰
Fig. 7. Rules of the getting-up alarm service

```
Problems  Javadoc  Declaration  Console  Git Staging
<terminated> RuleDerivation [Java Application] C:₩Program Files₩Java₩jre1.8.0_77₩bin₩javaw.exe (2017. 9. 29. 오후 5:41:47
Model location: IotOnt.owl
Rules location: alarm.rules
Rule bed_used concluded (iot:bed iot:used 'true'^^http://www.w3.org/2000/01/rdf-schema#Literal) <-
    Fact (iot:bed_pressure_sensor iot:belongTo iot:bed)
    Fact (iot:bed_pressure_sensor iot:hasValue '45.0'^^http://www.w3.org/2001/XMLSchema#double)
```

그림 8. 기상 알람 서비스 룰 기반 추론
Fig. 8. Inference based on the rules of the getting-up alarm service

그림 7은 구현한 온톨로지를 사용하여 스마트 홈 기상 알람 서비스를 기술한 Jena 룰을 보여준다. 스마트 홈 기상 알람 서비스는 bed_used, smart_alarm이라는 두 가지 룰로 기술한다. bed_used 룰은 bed_pressure_sensor의 값이 20.0보다 크다면 bed_pressure_sensor가 소속(belongTo) 되어 있는 PhyscicalObject 클래스의 인스턴스는 사용 중인 상태

를 갖게 된다. 즉, (bed_pressure_sensor, belongTo, bed) 이고, (bed_pressure_sensor, hasValue, 45.0) 이면, bed_pressure_sensor의 값이 20.0 보다 크므로, (bed, used, "true")를 추론하게 된다. 그림 8은 그림 6에서 사용한 데이터 모델(IotOnt.owl)과 그림 7의 홈 기상 알람 서비스 룰(alarm.rules)을 입력 받았을 때, Jena의 RuleReasoner를 사용하는 RuleDerivation 프로그램이 bed_used 룰의 조건이 만족되어 (bed, used, "true")가 추론된 결과를 보여준다.

그림 7의 smart_alarm 룰은 kim이 bedroom에 위치해 있고, bedroom에 있는 bed가 사용 중이고, bed_lamp가 켜져 있고, 현재 시간이 오전 5시이면, tv을 켜고, kim_smartPhone의 알람을 켜고, curtain을 올리고, hanging_lamp을 켠다로 기술한다.

그림 9는 스마트 홈 방범 서비스 시나리오를 구현한 온톨로지에서 관련 주요 상황정보를 SPARQL을 사용하여 검색한 결과이다. kim_business_trip 스케쥴의 시작 시간(from)은 2017년 7월 24일 9시 부터이고, 종료 시간(to)은 2017년 7월 28일 19시까지이다. 현재(current) 시간(dateTime)이 2017년 7월 25일 19시이고, curtain은 올라가 있는(up) 상태이고, hanging_lamp는 꺼져(off) 있다는 것을 나타낸다. 따라서 그림 9의 질의 결과는 구현한 온톨로지가 스마트 홈 방범 서비스의 상황정보를 잘 표현한다는 것을 나타내고 있다.

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX iot: <http://www.semanticweb.org/seoil/ontologies/2017/7/IotOnt#>
SELECT ?subject ?object
        WHERE {{?subject rdf:type iot:Schedule; iot:from ?object.} union
        {?subject rdf:type iot:Schedule; iot:to ?object.} union
        {?subject rdf:type iot:Person; iot:locatedIn ?object.} union
        {?subject rdf:type iot:Time; iot:dateTime ?object.} union
        {?subject rdf:type iot:Furnishing; iot:status ?object.} union
        {?subject rdf:type iot:Device; iot:power ?object.}}
```

subject	object
kim_business_trip	"2017-07-24T09:00:00"^^<http://www.w3.org/2001/XMLSchema#dateTime>
kim_business_trip	"2017-07-28T19:00:00"^^<http://www.w3.org/2001/XMLSchema#dateTime>
current	"2017-07-25T19:00:00"^^<http://www.w3.org/2001/XMLSchema#dateTime>
curtain	"up"^^<http://www.w3.org/2000/01/rdf-schema#Literal>
tv	"off"^^<http://www.w3.org/2000/01/rdf-schema#Literal>
bed_lamp	"off"^^<http://www.w3.org/2000/01/rdf-schema#Literal>
hanging_lamp	"off"^^<http://www.w3.org/2000/01/rdf-schema#Literal>

그림 9. 방범 서비스 상황정보 온톨로지 검색
Fig. 9. Situation information ontology retrieval for security service

```
@prefix  iot: <http://www.semanticweb.org/seoil/ontologies/2017/7/IotOnt#>.

[evening: (iot:kim_business_trip iot:from ?from),
          (iot:kim_business_trip iot:to ?to),
          (iot:current iot:dateTime ?x),  ge(?x, ?from), le(?x, ?to),
          (iot:current iot:time "19:00:00"^^rdfs:Literal)
          -> (iot:curtain iot:status iot:"down"^^rdfs:Literal ),
             (iot:hanging_lamp iot:power "on"^^rdfs:Literal) ]

[night:   (iot:kim_business_trip iot:from ?from),
          (iot:kim_business_trip iot:to ?to),
          (iot:current iot:dateTime ?x),  ge(?x, ?from), le(?x, ?to),
          (iot:current iot:time "23:00:00"^^rdfs:Literal)
          -> (iot:hanging_lamp iot:power "off"^^rdfs:Literal) ]

[morning: (iot:kim_business_trip iot:from ?from),
          (iot:kim_business_trip iot:to ?to),
          (iot:current iot:dateTime ?x),  ge(?x, ?from), le(?x, ?to),
          (iot:current iot:time "09:00:00"^^rdfs:Literal)
          -> (iot:curtain iot:status iot:"up"^^rdfs:Literal )]
```

그림 10. 방범 서비스 룰
Fig. 10. Rules of the security service

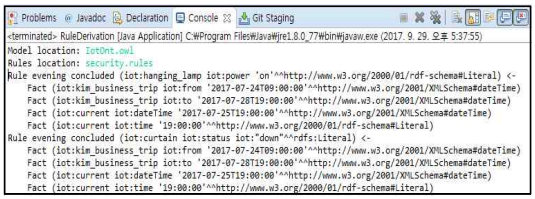


그림 11. 방범 서비스 룰 기반 추론
Fig. 11. Inference based on the rules of the security service

그림 10는 스마트 홈 방범 서비스를 기술하는 Jena 룰을 보여준다. 스마트 홈 방범 서비스는 evening, night, morning 이라는 세 가지 룰로 기술된다. evening, night, morning 룰 모두 공통적으로 현재(current) 시간(dateTime)이 kim_business_trip의 시작 시간(from) 보다 크거나 같고, kim_business_trip의 종료 시간(to) 보다 작거나 같다는 조건을 만족해야 한다. evening 룰은 19시 일 때, 커튼을 내리고(down), hanging_lamp을 켠다(on). night 룰은 현재 시간이 23시 일 때, hanging_lamp을 끈다(off). Morning 룰은 현재 시간이 9시 일 때, 커튼을 올린다(up). 그림 11은 그림 9의 데이터 모델(IotOnt.owl)과 그림 10의 스마트 홈 방범 서비스 룰(security.rules)를 입력받아 evening 룰의 조건이 만족되었을 때, (hanging_lamp, power, "on")과 (curtain, status, "down")이 Jena의 RuleReasoner를 사용하는 RuleDerivation 프로그램에 의

해 추론된 결과를 보여준다.

그림 12는 Schedule의 인스턴스인 kim_business_trip 인스턴스의 생성시간과 생존시간의 메타데이타를 구현한 owl의 xml 인코딩을 보여준다. kim_business_trip 인스턴스의 type은 Schedule이고, from 속성과 to 속성을 갖는다. 생존시간을 나타내는 lifeTime annotation 속성의 값으로는 3600이라는 값(초 단위)을 갖고, 생성 시간을 나타내는 productionTime annotation 속성의 값으로는 2017-07-17T09:00:00 이라는 dateTime 자료형 데이터를 갖는다는 것을 보여준다. 그림 13은 메타데이타를 구현한 온톨로지에서 kim_business_trip 인스턴스의 메타데이타를 SPARQL을 사용하여 검색한 결과를 보여준다. 검색한 결과는 상황정보 수집기에서 상황정보를 수집하기 위해서 사용한다.

```
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/seoil/ontologies/2017/7/IotOnt#kim_business_trip">
    <rdf:type rdf:resource="http://www.semanticweb.org/seoil/ontologies/2017/7/IotOnt#Schedule"/>
    <from rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2017-07-24T09:00:00</from>
    <to rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2017-07-28T19:00:00</to>
    <lifeTime rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3600</lifeTime>
    <productionTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2017-07-17T09:00:00</productionTime>
</owl:NamedIndividual>
```

그림 12. 'kim_business_trip'의 메타데이타
Fig. 12. Metadata of 'kim_business_trip'

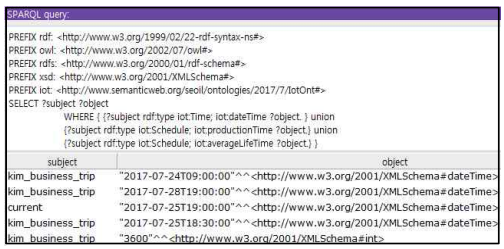


그림 13. 'kim_business_trip' 메타데이타 검색
Fig. 13. 'kim_business_trip' metadata retrieval

5. 결론

IoT 환경의 상황정보를 의미적인 상호운용성을 지원할 수 있도록 표현하기 위하여 본 논문에서는 온톨로지 기반의 상황정보 모델을 제안하였다. 제안한 모델은 온톨로지 언어인 OWL을 사용하여 기술하였으며, 다양한 도메인의 상황 정보를 모델링할 수 있도록 상위 온톨로지를 구축하였다. IoT 환경을 구성하는 사람, 환경, 사물 등을 클래스로 표현하고 그 상황을 나타내는 속성을 정의하여 일관성 있게 상황 정보를 기술하였기 때문에 이해하기 쉽고 사용하기 쉽다. IoT 서비스 정보를 중심으로 상황정보를 온톨로지로 정의한 선행 연구들은 온톨로지 모델이 이해하기 어렵고 복잡하다는 단점이 있다. 본 논문에서 IoT 환경의 서비스는 구축한 온톨로지 모델을 기반으로 Jena 룰을 사용하여 기술함으로써 복잡한 서비스를 기술할 수 있도록 하였다. 또한 상황정보 메타데이타 개념을 적용하여 상황정보의 유효성을 판단할 수 있는 온톨로지를 정의한 장점이 있다.

본 논문에서 제안한 온톨로지 기반의 IoT 환경의 상황정보 모델은 향후, IoT 상황 정보 생성기, 수집기 및 서비스를 제공하는 어플리케이션 실행 환경을 구축할 때, IoT 시스템의 각 구성 요소들이 상황 정보 모델을 공유하여 상호운영성을 제공할 수 있게 한다.

REFERENCES

[1] Yoo, Jinho, "The IoT Implementation Technology for e-Health Device Connection," Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 8, No. 5, pp. 394~399, Oct, 2015.
[2] T. R. Gruber, "A Translation Approach to Portable Ontologies", Knowledge Acquisition Journal, Vol. 5, pp. 199-220, 1993.
[3] OWL, Available at: https://www.w3.org/OWL/
[4] Nari Yang, Hoan-Suk Choi and Woo-Seop Rhee, "Development of the Cross-vertical Ontology for Context Aware Service in Various IoT Environment", JOURNAL OF THE KOREA CONTENTS ASSOCIATION, Vol. 15, No. 2, pp. 58~73. 2015,
[5] Dong Hee Woo, Min Kyu Yoo and Yoon

Ho Kim, "A Study on Ontology for Semantic-Based Service Exploiting the Context Information in IoT Environment," The Jounal of Society for e-Business Studies, Vol. 21, No. 3, pp. 1~13. 2016,
[6] Je-Min Kim, Mi-Hwa Kim and Young-Tack Park, "MOnCa : Framework for Ontology-based Context Aware Smart Phone Applications," Journal of KISS : Software and Applications, Vol. 38, No. 7, pp. 369~381. 2011,
[7] Robert Hoehndorf, "What is an upper level ontology?", Available at: http://ontogenesis.knowledgeblog.org/740, April 13, 2010.
[8] Bae, Hong-Min, Seo, Shin-Il and Kim, Byung-Seo, "Home Automation System through Learning User Life Pattern," Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 8, No. 2, pp. 79~85, Apr, 2015.
[9] Eunhoe Kim, Jaeyoung Choi, "An Ontology-based Context Model in s Smart Home", ICCSA 2006, LNCS 3983, pp 11-20, May 2006.
[10] Protege, Available at: https://protege.stanford.edu/
[11] Apache Jena, Available at: https://jena.apache.org
[12] SPARQL Query Language, Available at: https://www.w3.org/TR/rdf-sparql-query/

저자약력

김 은 희(Eunhoe Kim) [정회원]

- 1998년 8월 : 숭실대학교 컴퓨터학과 (공학석사)
- 2006년 8월 : 숭실대학교 컴퓨터학과 (공학박사)
- 2007년 8월 ~ 2012년 2월 : 숭실대학교 정보미디어기술연구소, 지능형로봇연구소 전임연구원
- 2013년 3월 ~ 현재 : 서일대학교 소프트웨어공학과 조교수

<관심분야> IoT, 분산처리, 클라우드 컴퓨팅

서 유 화(Yuhwa Suh) [정회원]

- 2005년 8월 : 숭실대학교 컴퓨터학과 (공학석사)
- 2016년 2월 : 숭실대학교 컴퓨터학과 (공학박사)
- 2007년 11월 ~ 2009년 10월 : 정보통신연구진흥원((현)정보통신산업진흥원) 연구원
- 2016년 3월 ~ 현재 : 서일대학교 정보통신공학과 조교수

<관심분야> 그린네트워킹, 유무선통신, 네트워크보안