

# IoT 스마트 홈 환경을 위한 상황 인식 추론 프레임워크 설계 및 구현

이정준<sup>○</sup>, 김정태\*, 윤희용\*

<sup>○\*</sup>성균관대학교 정보 통신대학

e-mail: jungjune86@skku.edu<sup>○</sup>, kyungtaekim76@gmail.com\*, youn7147@skku.edu\*

## Design and Implementation of Context-aware Inference Framework for IoT Smart Home Environment

Jung June Lee<sup>○</sup>, Kyung Tae Kim\*, Hee Yong Youn\*

<sup>○\*</sup>College of Information and Communication Engineering, Sungkyunkwan University

### ● 요약 ●

과거 유비쿼터스 기술의 출현 이후로 사물에 간단한 인식 센서를 이용한 형태의 서비스가 널리 보급되었고, 스마트 기기의 발달로 인해 PC가 아닌 환경에서도 인터넷을 사용하기 용이한 환경이 정착되어, 이들을 이용한 사물 인터넷 (Internet of Things) 환경이 빠르게 확산 중이다. 본 논문에서는 상황 인식 서비스와 추론 서비스를 사물 인터넷 환경에 적용 시킨 스마트 홈 상황 인식 추론 프레임 워크의 설계 및 구현을 서술한다. 해당 프레임 워크는 실질적인 상황 정보를 제공하는 이기종의 사물 인터넷 기기 간 데이터 타입을 수용하기 위해 온톨로지 언어인 OWL 규격을 사용하여 상황 정보를 수용하고, 룰 입력 모듈을 통해 다양한 환경을 모델링 할 수 있는 XML 규격의 서비스 룰을 입력받는다. 이후, 상황 정보와 서비스 룰을 기반으로 추론엔진을 통해 상황을 추론하여, 단순히 조건 만족 시 실행 구조가 아닌 상황 기반의 추론에 의한 서비스를 제공하게 된다. 프레임 워크를 활용 방안을 설명하기 위해 예제 방법 시나리오를 통해 해당 프레임 워크의 특징 및 서비스의 흐름을 서술한다.

키워드: 사물 인터넷(Internet of Things), 상황 인식(Context Aware), 추론 엔진(Inference engine)

## I. 서론

과거 유비쿼터스 기술의 출현 이후로 사물에 간단한 인식 센서를 이용한 형태의 서비스가 널리 보급되었고, 스마트 기기의 발달로 인해 PC가 아닌 환경에서도 인터넷을 사용하기 용이한 환경이 정착되었다. 또한 클라우드, 빅 데이터 등 대용량 데이터 관리 개념이 발전하면서 기존의 기술과 융합되어, 모든 사물이 인터넷 통신 및 연산을 할 수 있는 IoT(Internet of Things) 환경이 빠르게 확산 중이다. 이에 맞추어 기존의 유비쿼터스, PC, 웹 등에서 제공되던 서비스들이 IoT 환경에 적용하여 더욱 다양한 서비스를 제공하기 시작했다. 본 논문에서는 상황 인식 서비스와 추론서비스를 IoT 환경에 적용 시킨 스마트 홈 상황인식 추론 서비스 구현을 서술한다. 논문의 구성 다음과 같다. 2장에서 프레임워크를 이루는 상황 인식에 대한 개념과 사물 인터넷 기기들이 전송하는 데이터 포맷의 통일을 위해 사용한 OWL의 개념, 추론 엔진의 개념 및 알고리즘에 대해 간략히 서술한다. 3장에서는 프레임워크의 설계와 예제 시나리오를 통해 2장의 개념들을 어떤 식으로 융합했는지 서술하고, 이 시스템의 활용 방안에 대한 예시를 든다. 마지막으로 4장에서 결론 및 향후 연구 방향을 기술한다.

## II. 관련 연구

### 1. 상황 인식

상황인식의 개념은 1994년에 Schilit 과 Theimer 가 최초로 사용하였고[1], 지금까지 상황 인식에 대해 다양한 각도로 개념이 정의되었다.[2] 어플리케이션의 개발관점에서 상황인식에 대한 정의는 다음과 같다.

정의 1 : 상황 정보란 어떤 개체의 상태를 나타내는 모든 정보이다.

개체는 사람, 장소, 물체 등 사용자와 어플리케이션 사이에 상호작용을 하는데 있어서 고려해야 할 요소들을 의미한다.

정의 2. : 시스템이 작동함에 있어 사용자에게 상황 정보와 관련된 정보를 제공 할 경우 해당 시스템은 상황 인식 시스템이라 할 수 있다.

이러한 상황인식 서비스의 대표적인 예로는 헬스 케어와 방법 시스템이 있다.

### 2. OWL

OWL은 정보를 표현하는 방법으로, 2004년에 W3C의 권고안으로 제정되었다[3]. OWL은 XML, RDF, RDFS 의 언어와 비교하여 더욱 풍부한 어휘(vocabulary) 와 형식적 의미론(formal semantics)

를 포함하고 있어서 컴퓨터가 정보를 해석함에 있어 기존의 언어보다 뛰어난 성능을 보인다. OWL은 표현 능력에 따라 OWL Lite, OWL DL, OWL Full 로 구분되며, 후자로 갈수록 더 다양한 정보의 표현이 가능하지만 복잡도는 증가한다. OWL의 구분은 XML문서 선언문, 문서의 형태 선언문, Namespaces, 온톨로지 헤더(Header), OWL의 구분에는 클래스, 속성, 각 속성간의 관계 등의 구분들이 포함된다.

### 3. 추론 엔진

추론 엔진은 기존의 지식 정보에서 추론 룰을 이용하여 새로운 지식 정보를 추론 하는 소프트웨어 도구이다[3]. 그림 1. 은 추론엔진 기법중 하나인 전향 추론 에 대한 개념도이다.

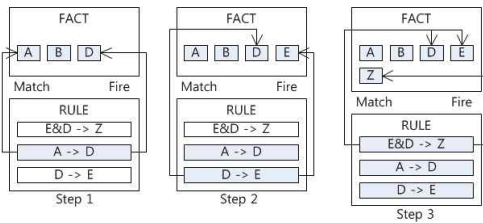


그림 1. 전향 추론

Fig. 1. Forward Chaining

추론 엔진은 크게 룰과 팩트로 이루어져 있다. 룰은 조건과 실행으로 구성되며 팩트와 룰의 조건부를 비교하여 동일 할 경우 실행부분을 구동 하게 된다. 전향 추론은 이러한 룰 과 팩트의 비교 방법 중 하나로, 원인에서 결과를 유추하는 방법론이다. 그림 1은 최초 A,B 팩트만 있는 상황에서 첫 단계로 룰 A->D 와 A 팩트를 비교하여 팩트 D가 추가되며, 이리인해 두 번째 단계인 룰 D->E 와 추가된 팩트 D와의 비교를 통해 팩트 E의 추가 후, 마지막으로 룰 E&D->Z와 팩트 E,D의 비교를 통해 Z를 얻는 과정을 나타낸다. 전향 추론은 모든 룰의 매칭 이후 더 이상 비교할 룰이 없거나, 사전에 제한한 매칭 횟수 만큼 매칭 시도가 끝날 경우 종료되며, 마지막에 매칭된 룰의 실행 부분이 해당 전향 추론의 결론이 된다. 전향 추론 외에 다양한 추론 기법이 존재하며, 대표적인 추론 엔진으로는 JESS[4] 가 있다.

## III. 본 론

본 논문에서는 기존의 상황 인식 서비스와 추론 엔진을 IoT 스마트 홈 환경에 접목 시켜서 상황과 추론에 의해 서로 다른 서비스가 상호 작용을 할 수 있는 프레임 워크를 제시한다. 해당 프레임 워크의 설계는 그림 2. 와 같다.

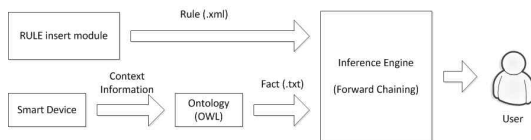


그림 2. 상황 인식 추론 프레임워크 설계도

Fig. 2. Context-aware Inference framework Architecture

Rule 입력 모듈에 의해 사용자에게 입력받은 룰들은 추론 엔진의 메모리에 적체 된다. 이후, 스마트 기기에 의해 발생한 상황 정보가 온톨로지를 이용한 OWL 포맷으로 일관되게 변환 시켜서 입력되면 추론 엔진에서 이를 팩트의 인자 값으로 이용 하여 전향 추론기법에 의해 추론된 결과를 사용자에게 제공한다. 스마트 홈 환경에서 주요한 이슈로는 사용자의 요구 및 다양한 기기의 데이터를 처리 할 수 있는 유동적인 데이터 처리 이며, 이러한 이슈 해결을 위해 룰 입력과 디바이스의 정보 입력을 표준 포맷인 XML[5]과 OWL[6]을 사용 하였다. 그림 3. 은 룰 입력 시 생성되는 XML 문서의 구조를 나타낸다.

```
<Activity Id="rule_1" Name="check window">
  <Rule>Window movement</Rule>
</Activity>
<Activity Id="excute_1" Name="warning">
  <Subject>alarm</Subject>
  <Predicate>on</Predicate>
</Activity>
```

그림 3. RULE XML 문서 구조

Fig. 3. RULE XML Format

룰 입력 모듈을 이용하여 사용자가 룰의 조건 부분을 입력하면 XML문서의 Rule 노드에 조건부의 데이터가 생성되고, 사용자가 룰의 실행 부분을 입력하면 XML문서의 Subject 노드와 Predicate 노드에 실행 데이터가 나뉘어서 입력된다. 이 XML 문서는 시스템에 의해 파싱되어 조건부와 실행부의 데이터만 룰 정보로 사용된다. 사용자의 입력에 의한 XML 데이터 생성이 자동화 되어 있으므로 사용자가 어떠한 룰을 입력하더라도 시스템이 체계적으로 인식 할 수 있는 장점이 있다.

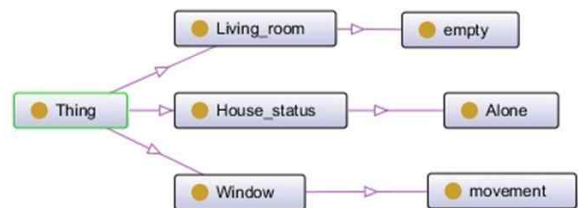


그림 4. 상황 정보 개념도

Fig. 4. Context Information

그림 4. 는 각각의 스마트 디바이스에서 입력되는 상황 정보를 나타낸다. 사용자가 어떠한 스마트 디바이스를 스마트 홈 환경에 추가 시킬지는 알 수 없으며, 이는 각 장비마다 제공하는 데이터의 포맷이 다를 수 있음을 의미한다. 본 시스템에서는 이러한 다양한 포맷의 데이터를 수용하고자, 정보를 표현하는 표준 규격인 OWL 데이터 포맷을 사용하였다. 각 장비에서 제공하는 데이터가 표준 규격인 OWL 의 형태로 시스템에 전송되면, 시스템은 OWL 데이터를 분석하여 이를 추론 엔진의 팩트 인자 값으로 활용한다. 룰 입력 모듈을 통해 입력받은 룰 데이터와 OWL을 이용하여 통일시킨 상황 정보를 이용하여, 추론엔진은 사용자에게 상황에 맞는 결과를 처리하

여 제공 한다.

이와 같은 상황 인식 추론 프레임워크에서 방법시스템을 운영 할 경우, 표 1, 표2와 같은 데이터가 입력되었을 때 그림 5. 와 같은 순서로 실행이 된다.

표 1. 방법 시스템 룰 정보

Table 1. Security System Rule Information

룰	조건	실행
1	창문에서 움직임이 감지됨	알람 발생
2	거실이 비어있음 & 알람발생	도둑 침입
3	도둑 침입 & 집에 혼자 있음	사용자에게 경찰서 연락처 정보 제공

표 2. 방법 시스템 상황 정보

Table 2. Security System Context Information

팩트	상황 정보
1	집에 혼자 있음
2	거실이 비어있음
3	창문에서 움직임이 감지됨

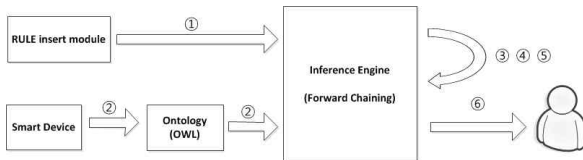


그림 5. 스마트 홈 방범 시나리오

Fig. 5. Smart Home Security Scenario

그림 5. 의 실행 순서에 따른 동작은 표 3 과 같다.

표 3. 스마트 홈 방범 시나리오 동작 순서

Table 3. Smart Home Security Scenario Order

순서	설명
1	사용자가 표 1 에 있는 3가지 룰을 입력한다.
2	각 스마트 디바이스에서 표 2 와 같은 상황 정보를 입력받는다.
3	팩트 3과 룰 1을 비교하여 같음을 확인 한 후, 알람 발생을 실행한다. 이후 알람 발생은 팩트에 추가로 입력된다.
4	3에서 추가된 알람 발생 과 기존의 팩트 2를 룰 2와 비교하여 같음을 확인한다. 이후 도둑 침입 상태를 활성화 시키고, 해당 실행 동작은 팩트에 추가로 입력된다.
5	4에서 추가된 도둑 침입 과 기존의 팩트 1을 룰 3과 비교하여 같음을 확인한다. 이후 사용자에게 경찰서 정보를 제공하는 동작이 실행된다.
6	5에서 모든 룰이 실행되었으므로, 룰 3 의 실행부를 결론이라 판단한다. 이후 사용자에게 추론의 결과인 경찰서 연락처 정보를 제공한다.

해당 시스템의 상황 정보 중 팩트 2: 거실이 비어있음 항목이 거실이 비어있지 않음 일 경우 룰 2 의 조건이 만족되지 않으므로 도둑 침입 과 경찰서 정보 제공을 발생하지 않게 되며 이는 시스템이 상황정보를 보다 유연하게 활용하여 대응 할 수 있음을 의미한다. 이와 같은 시나리오의 실행 과정을 통해 해당 시스템은 단순히 조건 만족 시 실행을 하는 구조가 아닌 상황 정보를 이용한 추론을 통해 동작함을 알 수 있다. 시스템의 실행 결과는 아래 그림 6. 과 같다.

```

Inference engine: compare RULE [alarm on null] FACT [alarm on null] MATCHED!
Inference engine: Postcondition of RULE executed!! [thief enter null]
Inference engine: New FACT added! [thief enter null]
Inference engine: compare RULE [alarm on null] FACT [thief enter null]
Inference engine: compare RULE [living_room empty null] FACT [class movement typest]
Inference engine: compare RULE [living_room empty null] FACT [window movement null]
Inference engine: compare RULE [living_room empty null] FACT [alarm on null]
Inference engine: compare RULE [thief enter null] FACT [thief enter null]
Inference engine: compare RULE [thief enter null] FACT [Ontology Null typest]
Inference engine: compare RULE [thief enter null] FACT [Class Alone typest]
Inference engine: compare RULE [thief enter null] FACT [House status Alone null]
Inference engine: compare RULE [thief enter null] FACT [Class House status typest]
Inference engine: compare RULE [thief enter null] FACT [Class living_room typest]
Inference engine: compare RULE [thief enter null] FACT [Class movement typest]
Inference engine: compare RULE [thief enter null] FACT [Class window typest]
Inference engine: compare RULE [thief enter null] FACT [alarm on null]
Inference engine: compare RULE [thief enter null] FACT [thief enter null]
Inference engine: compare RULE [thief enter null] FACT [thief enter null] MATCHED!
Inference engine: compare RULE [house_status Alone null] FACT [Ontology Null typest]
Inference engine: compare RULE [house_status Alone null] FACT [Class Alone typest]
Inference engine: compare RULE [house_status Alone null] FACT [Class Alone typest]
Inference engine: compare RULE [house_status Alone null] FACT [house_status Alone null] MATCHED!
Inference engine: Postcondition of RULE executed!! [police call null]
    
```

그림 6. 스마트 홈 방범 시나리오 실행 결과

Fig. 6. Smart Home Security Scenario Result

그림 6은 시스템 실행의 로그 화면으로, 추론 엔진이 룰과 팩트를 비교해서 매칭하고, 최종적으로 사용자에게 추론의 결과인 경찰서 정보를 제공 하는 모습을 나타낸다. 이와 같은 시스템 시나리오를 통해 해당 프레임 워크가 방범 분야에서 활용 될 수 있음을 알 수 있으며, 룰 정보를 도메인에 알맞게 적용할 경우 헬스 케어 등의 다양한 분야에서 활용 할 수 있음을 증명하였다.

#### IV. 결론

본 논문에서는 상황 인식 서비스와 추론서비스를 IoT 환경에 적용 시킨 스마트 홈 상황인식 추론 서비스 구현을 서술한다. 실질적인 상황 정보를 제공하는 이가중의 IoT 기기 간 데이터 타입을 수용하기 위해 온톨로지 언어인 OWL 규격을 사용하여 상황 정보를 수용하고, 룰 입력 모듈을 통해 다양한 환경을 모델링 할 수 있는 XML 규격의 서비스 룰을 입력받는다. 이후, 상황 정보와 서비스 룰을 기반으로 추론엔진을 통해 상황을 추론하여, 단순히 조건 만족 시 실행 구조가 아닌 상황 기반의 추론에 의한 서비스를 제공한다. 또한 해당 시스템을 이용해 방법 시나리오를 실행 함으로써 본 논문이 제시한 시스템이 정상적으로 상황 정보를 인식하여 추론함을 증명하였다.

향후 연구방향으로 서비스 디스커버리 기능을 추가하여, 해당 프레임워크에서 필요한 기능을 디스커버리 기능을 이용하여 찾을 수 있도록 개선 할 예정이다.

#### ACKNOWLEDGMENT

본 연구는 BK21+사업, 한국연구재단 기초연구사업 (2013R1A1A2040257), (2013R1A1A2060398), 삼성전자(S-2014-0700-000), 미래부가 지원 한 2013 년 정보통신-방송(ICT) 연구개발 사업 (1391105003)의 지원을 받아 수행되었음.

## 참고문헌

- [1] B. N. Schilit and M. M. Theimer, "Disseminating Active Map Information to Mobile Hosts", IEEE Network, Vol. 8, No. 5, pp. 22-32, 1994.
- [2] A. K. Dey, G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness", In the Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), 2000.
- [3] Deborah L. McGuinness, et al, "OWL Web Ontology Language Overview (W3C Recommendation 10 February 2004)", Available in:  
<http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [4] Swapna Singh, Ragini Karwayun "A Comparative Study Of Inference Engines", Seventh International Conference on Information Technology, pp. 53-57, 2010
- [5] Arun N Nambiar, Anish.K. Dutta, "Expert System for Student advising using JESS", International Conference on Education and Information Technology, Pp. V1 312- V1 315, 2010
- [6] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau "Extensible Markup Language" W3C Recommendation 26 November 2008