

RDF 형태의 상황정보 제공을 위한 센서 데이터 변환 방법

박유상, 조용성, 최종선, 최재영
숭실대학교 컴퓨터학부

E-Mail: {yoosangpark, yongseong.cho, jongsun.choi, choi}@ssu.ac.kr

A sensor data converting method for providing RDF-based situational information

Yoosang Park, Yongseong Cho, Jongsun Choi, Jaeyoung Choi
Dept. of Computer Science, Soong-Sil University

요 약

상황인지는 유비쿼터스 컴퓨팅 환경에서 사용자의 주변 상황을 인지하여 사용자가 원하는 서비스를 제공하기 위해 필요한 핵심 기술이다. 이러한 상황인지를 위해 여러 센서로부터 발생하는 저수준의 컨텍스트 정보를 처리하는 다양한 방법들이 존재한다. 그러나 현재 상황인지 처리에 관련된 표준 방법이 없어 서비스 도메인에 제한되고 복잡한 구현방법을 따라야 하며, 상황정보를 처리하는 시스템에 정형화된 상황정보를 제공하는데 어려움이 있다. 이에 본 논문에서는 정형화된 상황정보를 제공하기 위한 센서 데이터 변환 방법을 제안한다. 제안하는 센서 데이터의 변환 방법은 센서로부터 발생하는 저수준의 컨텍스트를 RDF 기반의 고수준의 상황정보로 변환하며, 변환된 정보는 상황인지 시스템에 제공된다.

Keywords : Ubiquitous computing, Sensor data converting, Situational information, Context, RDF, 유비쿼터스 컴퓨팅, 센서 데이터 변환, 상황정보, 컨텍스트

1. 서론

유비쿼터스 컴퓨팅[1]은 컴퓨팅 및 네트워크 환경이 장소, 시간에 구애받지 않고 적용된 컴퓨팅 환경을 일컫는데, 사물에 스며든 컴퓨터가 인간의 삶을 운택하게 만들어야 한다는 Mark Weiser의 철학을 바탕으로 본격적인 연구가 시작되었다. 컴퓨터가 스며든다는 개념의 근본적인 의미는 장비의 소형화와 더불어 주변 환경의 변화에 따라 제공되는 서비스도 변해야 한다는 것으로, 상황인지[2,3]는 Mark Weiser의 주관을 반영하는 유비쿼터스 컴퓨팅의 핵심기술 중 하나다. 상황인지는 사용자 맞춤형 서비스를 위해 고수준의 상황정보가 필요한데, 이는 주변 상황정보를 가공하여 획득할 수 있다. 고수준 상황정보로 변환하는 방법에 대한 연구는 각 분야 시스템에서 필요한 특정 도메인을 위한 구조 설계 방법부터 추론을 기반으로 한 변환 방법 등 다양한 관점으로 진행되었다[4,5].

로봇 서비스는 상황인지를 기반으로 한 시스템의 대표적인 예로, 주변 환경정보를 인지하여 사용자에게 맞춤형 서비스를 제공해 준다. 제공할 서비스의 적합성 여부 판단은 고수준의 상황정보를 토대로 이뤄지고, 그때 센서 데이

터의 변환방법이 필요하다. 이에 대한 연구들은 신호처리 또는 추론에 초점을 맞춘 연구들이 다양하게 진행되었다.

본 논문은 온톨로지에 RDF 기반으로 정의된 센서 정보와 상황인지 워크플로우 언어를 바탕으로 센서로부터 발생하는 저수준의 컨텍스트를 고수준의 상황정보로 변환하는 방법을 제안한다.

2. 관련 연구

Terry Winograd[6]는 상황인지 컴퓨팅 환경을 구축할 때, 그 구조를 크게 3가지로 모델링 할 수 있다고 정리하였는데 그 3가지는 Widgets, Networked Services 그리고 Blackboards 로 분류한다. Widgets은 센서 장비들이 하드웨어에 직접 연결되어 컴퓨터가 신호만 수신하는 가장 오래된 모델이다. Networked Services는 Widgets에 인터넷 기반 유/무선 통신기술을 더한 모델이다. 마지막으로 Blackboards는 Widgets, Network services 모델들을 확장시킨 모델 이지만 위 두 모델과는 다른 처리구조 관점을 가지고 있는데, 이는 처리중심(process centric) 또는 연결기반(connection oriented)이 아닌 데이터 중심(data centric)의 관점을 가진다.

The Active Badge Location System[7]은 Blackboards

본 논문은 한국 산업통산자원부의 로봇산업융합핵심기술사업 프로그램 (No. 10048474)의 지원으로 수행되었습니다.

모델을 기반으로 만든 시스템으로, 소형 Active badges를 사용자들이 휴대하고, 각지에 흩어진 센서 데이터 수집서버가 이를 확인하여 중앙서버로 보내어 위치 현황을 보여 주는 것이 주요 목적이다.

SOPRANO[8]는 온톨로지(Ontology) 기반 추론을 이용하여 사용자에게 맞춤형 서비스를 제공할 수 있는 시스템에 대한 연구이다. 해당 시스템에 관한 연구 뿐만 아니라 DAEDALOS[9]와 COMANTO[10]를 바탕으로 다양한 연구가 진행 중이다.

온톨로지[11]는 특정 분야의 데이터 또는 지식개념을 기술하는 모델로, 시멘틱웹을 구현할 수 있는 도구이다. Individual로 정의되는 개념들을 명세하기 위한 OWL(Web Ontology Language)과 정의한 개념 간 관계를 기술할 수 있는 RDF(Resource Description Framework), 추론규칙을 정의하기 위한 SWRL(Semantic Web Rule Language) 등을 바탕으로 OWLAPI[12] 또는 Jena API[13] 등을 이용하여 구현한다. 온톨로지를 상황인지 시스템에 접목시켰을 때 얻는 장점은 세 가지가 있다. 첫 번째, 데이터 타입 정의를 자유롭게 할 수 있다. 두 번째, 상황정보를 기술하기 위한 기본적인 주어-술어-목적어의 표현이 가능하다. 세 번째, 기술된 표현을 바탕으로 추론을 수행할 수 있다.

CAWL(Context-Aware Workflow Language)[14]은 상황인지 서비스를 제공할 수 있는 시나리오 기반 워크플로우 언어이다. 사용자의 상황정보 및 사용자가 받고자 하는 서비스의 명세는 어휘(Vocabulary)단위로 구성된 고수준 상황정보를 RDF 기반의 주어-술어-목적어 형태로 기술하고, 전이조건 간에 논리식을 적용할 수 있어 복합 워크플로우 서비스의 표현이 가능하다.

현재까지 수행된 센서 데이터 변환 방법을 위한 관련 연구들은 센서 데이터를 고수준의 상황정보로 변환하기 위하여 특정 센서 군으로 장비의 범위를 한정하거나, 온톨로지 기반 추론에 의존적이다. 특히 온톨로지 기반 추론은 수행 시 많은 시스템자원이 필요하기 때문에 적시, 적소에 제공되어야 할 사용자 맞춤형 서비스에는 적합하지 않다.

3. 고수준 상황정보를 위한 센서 데이터 변환 방법

본 장은 센서로부터 발생하는 저수준의 컨텍스트를 C-AWL문서에 기술된 고수준의 상황정보로 변환하는 방법에 대하여 기술한다.

제안하는 변환 방법 과정은 세 단계로 진행된다.

- 1) 확인해야 할 상황정보를 CAWL Executor로부터 수신한다.
- 2) 1)로부터 수신한 상황정보에 명세된 주어와 동사에 맞는 목적어를 온톨로지서에서 탐색한다.
- 3) 2)로부터 얻은 주어와 타겟 목적어에 함께 명세된 센서 정보를 이용하여 센싱결과를 상황정보로 유지하며 상황인지 요청처리를 수행한다.

CAWL문서에 정의된 상황정보를 수신하면 온톨로지에 정의된 센서 정보를 탐색하여 상황정보 요청을 처리하기 위한 매칭테이블을 생성한다. 이후 상황인지에 대한 서비스 요청이 들어올 때, 생성된 매칭테이블을 확인하여 요청을 처리한다.

처리구조의 관점에서 볼 때, 제안하는 변환 방법은 표현 중심(description centric)의 관점을 가진다. 제안하는 변환 방법에서 고수준의 상황정보와 온톨로지는 RDF 기반으로 표현되기 때문에 관련연구에서 언급한 연결중심이나 처리 중심과는 다른 관점을 가진다.

제안하는 변환 방법은 Fig 1의 로봇서비스를 제공하는 시스템에서 점선 영역이 해당한다.

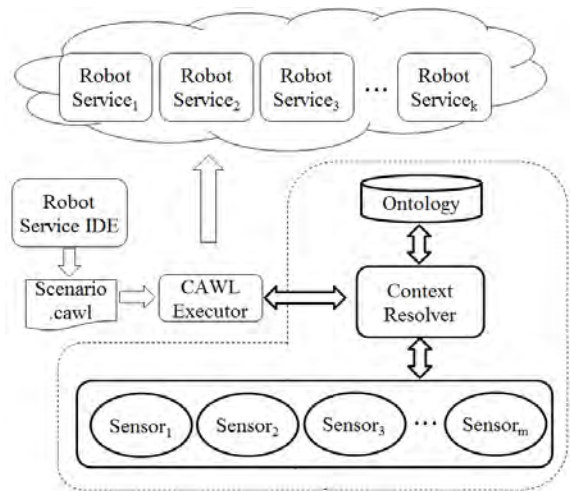


Fig 1. Structure of CAWL-based Robot service system

센서 데이터 변환을 위한 작업은 Context Resolver가 수행하고, Context Resolver는 온톨로지와 센서 정보를 이용한다. 이를 계층 구조로 살펴보면, Context Resolver와 온톨로지 및 센서 장비들은 아래 Fig 2와 같다.

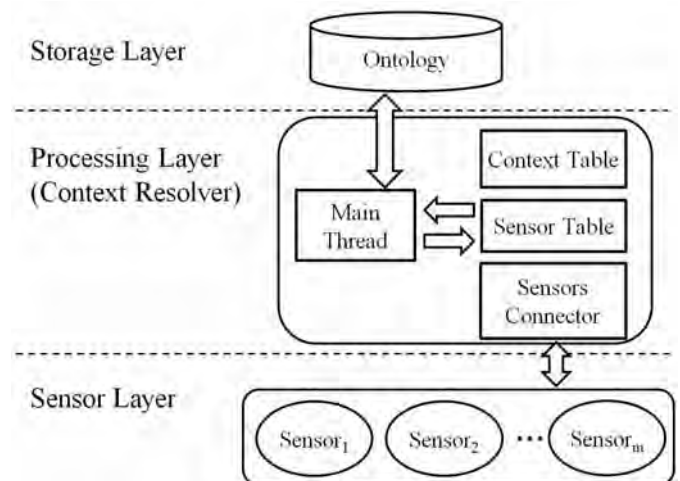


Fig 2. Layered Architecture for Converting sensor data

Context Resolver의 Main Thread는 Fig 1의 CAWL Executor로부터 Table 1과 같이 구성된 테이블을 수신하

고, 각 필드의 어휘는 온톨로지에 정의되어 있는 적합한 센서 정보를 찾는 과정에서 이용된다. 변환하는 과정은 Table 1의 Cid0에 정의된 상황정보를 예로 설명한다.

Table 1. Data structure for Context Resolver

CID	S(Subject)	V(Verb)	O(Object)
Cid0	Tom	isLocatedIn	Annex
Cid1	Ethan	isLocatedIn	MeetingRoomA
Cid2	MeetingRoomB	isHumidity	Damp
...

Cid0 상황정보를 센서로부터 확인하기 위한 절차는 Tom의 정보를 찾는 것부터 시작한다.

```

(A) RDF detail for subject
.
.
<owl:NamedIndividual rdf:about="&Context;Tom">
  <rdf:type rdf:resource="&Context;Person"/>
  <RFID>0C7CBF42</RFID>
  <PersonID>1124156</PersonID>
  <Entrance_Server>192.168.0.7</Entrance_Server>
  <Nickname>Tom</Nickname>
  <PersonName>Tom Smith</PersonName>
.
.
</owl:NamedIndividual>
.
.

(B) RDF detail for predicate
.
.
<owl:NamedIndividual
  rdf:about="&Context;isLocatedIn">
  <isConvertingTo rdf:resource="&Context;
  Location"/>
</owl:NamedIndividual>
.
.

(C) RDF detail for object
.
.
<owl:NamedIndividual rdf:about="&Context
;Sensor_Location_Annex">
<rdf:type rdf:resource="&Context;Location"/>
<MACAddress>00:2B:3F:BC:27:35</MACAddress>
<LimitValue>1</LimitValue>
<IP>192.168.0.64</IP>
<Nickname>Annex</Nickname>
<Module>RFID_Reader</Module>
<Module>Humidity</Module>
<Module>Temperature</Module>
.
.
</owl:NamedIndividual>
.
.
    
```

Fig 3. RDF representation of S-V-O individuals

주어(S) Tom의 정보를 온톨로지서 획득하기 위하여 Person 클래스에서 Fig 3-(A)과 같은 명세를 탐색한다. 술어(V) isLocatedIn은 Fig 3-(B)를 바탕으로 온톨로지 명세에 따라 Location으로 변환된다. isLocatedIn으로부터 변환된 Location은 목적어(O)를 탐색할 타겟 클래스가 되며, 해당 클래스에서 Fig 3-(C)와 같이 Nickname이 Annex로 정의된 명세를 찾는다.

위와 같이 단일 Cid에 대한 센서 정보를 모두 찾으려면

Main Thread는 획득한 정보를 Table 2, 3과 같이 구성된 테이블로 생성한다. Table 3은 ID와 주어(S)의 센서 정보 참조 값(Index), 이에 대응하는 센서로부터 센싱한 결과, 목적어(O)의 센서 정보 참조 값, 목적어에 대응하는 센서로부터 센싱한 결과, 최종상태 및 Timestamp로 구성된다.

모든 Cid에 대한 처리가 끝나면 완성된 테이블을 통하여 상황인지 판단 요청을 처리할 수 있다. 이는 Table 1.의 동사가 가지는 의미가 목적어에 포함 되어 있고, Context Table의 값(Index)을 바탕으로, Sensor Table의 값을 참조하여 필요한 센서를 확인하는 과정을 수행한다. 특정 Cid에 대한 상황인지 판단 요청을 처리 시에는, 각 주어와 목적어 센서들의 상태를 판단하고 조건을 만족하면 Context Table의 Timestamp에 마지막으로 확인한 시간을 기록한다.

Table 2. Data structure for Sensor table

Index	Keyword	Class	IP / ID	Status
0	Tom	Person	1124156	On
1	Annex	Location	192.168.0.64	On
2	Ethan	Person	1124157	Off
3	MeetingRoomA	Location	192.168.0.61	On
4	MeetingRoomB	Location	192.168.0.62	On
5	HumidSensor MeetingRoom_B	Humidity	192.168.0.112	On
...

Table 3. Data structure for Context table

ID	S	T/F	O	T/F	Status	Timestamp
Cid0	0	True	1	True	True	201404221321
Cid1	2	False	3	True	False	-
Cid2	4	True	5	True	True	201404221037
...

4. 실험

본 장은 제안한 방법을 테스트하기 위해서 어플리케이션을 구현하고 시뮬레이션을 통하여 설계한 구조대로 작동함을 확인한다.

어플리케이션은 다음의 언어로 구현하였다.

- Context Resolver : 자바 기반 어플리케이션
- Sensor Data : C 기반 통신전용 어플리케이션
- Ontology : Jena 2.11.0 API 기반 구축

Fig 4는 상황인지를 위한 어플리케이션의 진척도를 나타내는 좌측과 Context Table과 Sensor List를 나타내는 우측, 그리고 하단 콘솔화면으로 구성된다.

콘솔화면은 루틴별 수행되는 과정 등에 대한 로그가 출력된다. 각 센서는 Context Resolver의 요청에 대한 응답으로 정보를 전송해준다. 현재 과정은 “Tom isLocatedIn Annex”를 처리하고 타겟 센서들의 결과가 모두 True인 것을 확인하였기 때문에 Timestamp에 기록됨을 확인할 수 있다. Sensor Data 어플리케이션을 개별 목적을 가진

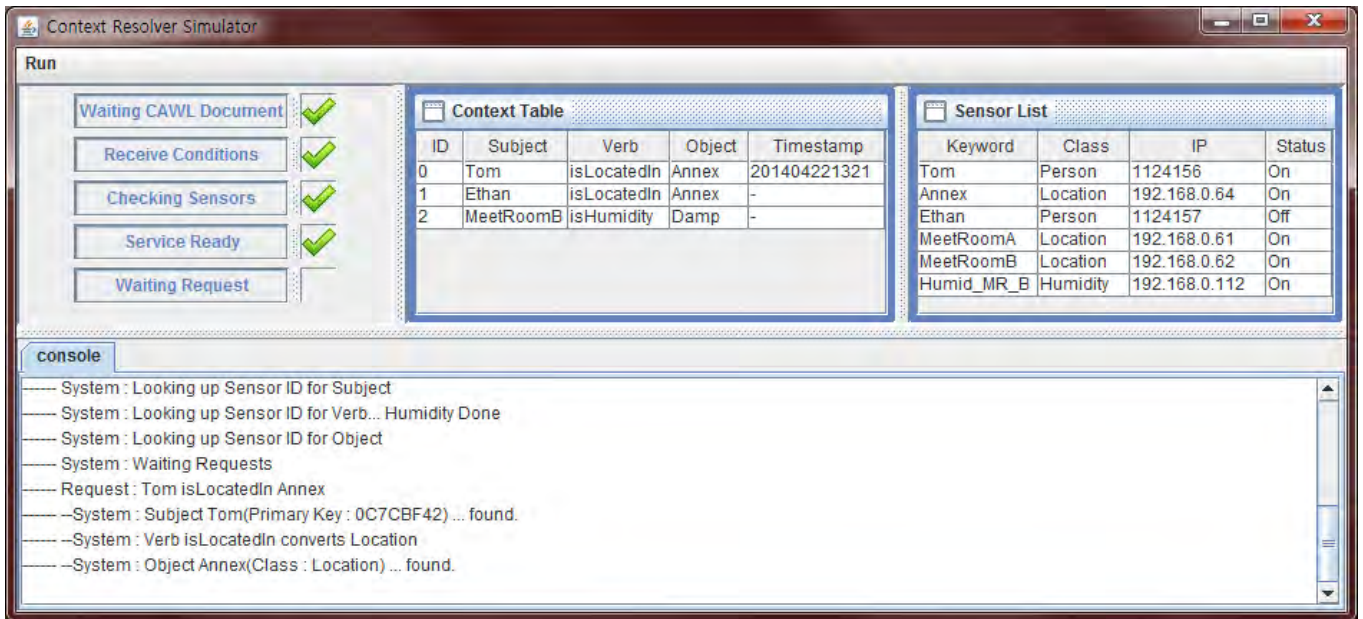


Fig 4. Context Resolver Simulator and Processing Cid0 “Tom isLocatedIn Annex”

센서 장비에 유/무선 통신모듈을 장착하여 대체하면 같은 결과를 실제 센서로부터 얻을 수 있을 것으로 기대된다.

5. 결론

본 논문에서 제안한 방법을 통하여 CAWL 문서에 기술된 고수준의 상황정보 및 온톨로지에서 획득한 센서 정보를 바탕으로 저수준의 컨텍스트를 고수준의 상황정보로 변환하였다. 변환된 고수준의 상황정보는 상황인지 기반 시스템에서 제공할 사용자 맞춤형 서비스를 제공하는데 도움을 준다.

한편, 제안한 센서 데이터 변환 방법은 각 센서의 모니터링을 할 수 있는 기능과 통신과정에서 발생할 수 있는 Fault-Tolerance 등을 처리할 수 있는 예외처리 방법이 추후 수행되어야 한다.

참고문헌

[1] M. Weiser, “The Computer for the 21st Century”, Scientific American, Vol 265, Issue 3, pp. 94-104, 1991
 [2] B. N. Schilit, M. M. Theimer, “Disseminating active map information to mobile hosts”, IEEE Network, Vol 8, Issue 5, pp. 22-32, 1994
 [3] A.K. Dey, “Understanding and using context”, Personal and ubiquitous computing, Vol 5, Issue 1, pp. 4-7, 2001
 [4] G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin. “Context-Aware Recommender Systems.” AI Magazine, vol. 32, no. 3, 2011
 [5] C. Henson, A. Sheth, K. Thirunarayan, “Semantic perception: Converting sensory observations to abstract-ions”, IEEE Internet Computing, Vol 16, Issue 2, pp. 26-34, 2012

[6] T. Winograd, “Architectures for context”, Human C-computer Interaction, Vol 16, Issue 2, pp. 401-409, 2001
 [7] R. Want, A. Hopper, “The active badge location system”, ACM TOIS, Vol 10, Issue 1, pp. 91-102, 1992
 [8] M. Klien, A. Schmidt, R. Lauer, “Ontology-centred design of an ambient middleware for assisted living: The case of soprano”, Artificial Intelligence: AIM-CU, KI 2007
 [9] DAIDALOS - <http://www.ist-daidalos.org/>
 [10] I. Roussaki, “Hybrid context modeling: a location-based scheme using ontologies”, PCCW Fourth Annual IEEE International Conference, pp. 6-7, 2006
 [11] D. McGuinness, F. Harmelen. “OWL web ontology language overview.” W3C recommendation 10.10, 2004
 [12] OWLAPI - <http://owlapi.sourceforge.net/>
 [13] Jena API - <https://jena.apache.org/>
 [14] Jongsun Choi, Yongyun Cho, Jaeyoung Choi, “The Design of a Context-Aware Workflow Language for Supporting Multiple Workflows”, KSII Transactions on Internet and Information Systems, Vol 10, Issue 6, 2009