Future Generation Computer Systems 33 (2014) 32–41

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An ontology-based approach to ADL recognition in smart homes ☆

Ihn-Han Bae *

School of Information Technology Engineering, Catholic University of Daegu, GyeongSan, GyeongBuk 712-702, Republic of Korea

HIGHLIGHTS

- We model the system architecture for RADL (Recognizing Activities of Daily Living).
- We design and implement the OWL-based ontology for RADL.
- We examine the rule reasoning for activity and service AmIApplications.
- The OWL-based ontology for RADL is verified and evaluated.
- RADL will improve the quality of life of elderly and disabled people.

ARTICLE INFO

Article history:
Received 8 May 2012
Received in revised form
28 March 2013
Accepted 6 April 2013
Available online 17 April 2013

Keywords:
Activities of daily living
Ontology
Smart home
Semantic reasoner
Web ontology language

ABSTRACT

This paper presents a method for recognition of Activities of Daily Living (ADL) in smart homes. Recognition of activities of daily living and tracking them can provide unprecedented opportunities for health monitoring and assisted living applications, especially for elderly people and people with memory deficits. This paper presents Recognizing Activities of Daily Living (RADL) by discovering and monitoring patterns of ADLs in sensor equipped smart homes. The RADL is composed of two components: smart home management monitoring and ADL pattern monitoring. This paper studies the ontology base and the reasoning that are the main parts of ADL pattern monitoring. The ontology for RADL is designed and the prototype system of RADL is implemented using Protégé and Jess tools. Also, the ontology for RADL is verified by OntoCheck in automatic mode and evaluated by a metric-based approach in manual mode.

1. Introduction

The convergence of healthcare system, sensors and communication technology is resulting in the development of a smart environment. All actions within the smart environment can be stored and analyzed. Ambient Intelligence (AmI) [1] in smart environments senses features of the users and their environment, then, reasons the accumulated data, and finally selects actions that will benefit the users in the environment. Smart homes designed under the AmI paradigm can improve the quality of life of elderly and disabled people by increasing the support received from the environment. Accordingly, our research focuses on the ADL [2–4] monitoring for smart homes under the AmI paradigm.

Smart homes [5–9], also known as automated homes, intelligent buildings, integrated home systems, are a recent design development. Smart homes incorporate common devices that control the features of the home. Originally, smart home technology was used to control environmental systems such as lighting and heating, but recently smart technology has evolved to an extent that almost any electrical component within a house can be included in the system. Moreover, smart home technology does not simply turn devices on and off; it can monitor the internal environment and the activities that are being undertaken whilst the house is occupied. As a result, a smart home can now monitor the activities of the occupant of a home and operate devices in set predefined patterns independently, as the user requires.

Recently, research on intelligent spaces that grasp actions of living people inside and help them has become active. Based on the liveliness, more and more information can be handled quickly by advanced information technology. We can gather and deal with data of life behaviors or changing surroundings as time series data based on various sensors. Various supports can be provided to users based on the time series data acquired by sensors. Some examples of such a support system include direct helps, such as carrying loads when getting home or preparing ingredients in meals, and indirect helps, such as batching our daily life and informing or warning in emergencies [10].

In this paper, we present RADL, a system that discovers and monitors patterns of ADLs in sensor equipped smart homes. The

☆ This work was supported by research grants from Catholic University of Daegu in 2011.

* Tel.: +82 538502742; fax: +82 538502750.
E-mail addresses: ihbae@cu.ac.kr, bae2742@hotmail.com.

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.04.004

RADL consists of two components: smart home management monitoring and ADL pattern monitoring. An ontology is an explicit, shared specification of the various conceptualizations in a problem domain. It defines commonly agreed data/knowledge structures, i.e., domain concepts, their attributes and the relations between them. Consequently, this paper presents the ontology base and the reasoning that are the main parts of ADL pattern monitoring. The ontology base supports the semantic discovery of location, device and environment domains in smart homes. The reasoning system discovers the activity of a person and the appropriate service for a present situation. On the detection of significant changes of context, the reasoning is triggered. The service manager of RADL invokes a home service or sends a message to remote family members according to the result of the reasoning. We design the ontology for RADL and implement the prototype system of RADL by using Protégé and Jess tools.

The remainder of this paper is organized as follows. Section 2 reviews related works on OWL ontology and smart homes. Section 3 designs the ontology for RADL by using the Protégé tool, and the ontology for RADL is verified through OntoCheck in automatic mode and is evaluated through a metric-based approach in manual mode. Section 4 presents the ADL reasoning process with Jess, OWL reasoning engine. Section 5 concludes the paper and discusses future work.

2. Background and related works

In this section, we present a description for ontology to form the knowledge base of RADL and describe the previous works for ADL recognition in smart homes.

2.1. OWL ontology

The term ''ontology'' originates from philosophy and refers to the discipline that deals with existence and the things that exist. Studer et al. [11] merged these two definitions stating that: ''An ontology is a formal, explicit specification of a shared conceptualization''. In computer science, an ontology is the standardized representation of knowledge as a set of concepts within a domain and the relationships between those concepts. It can be used to reason about the entities within that domain and can be used to describe the domain [12].

Typical elements of ontologies are (a) concepts and their attributes; (b) taxonomies to categorize concepts by generalization and specification; (c) relations between concepts; (d) axioms to define statements which are always true; and (e) individuals (or facts) that are instances of concepts and their relations [13].

Ontology languages allow users to write explicit, formal conceptualization of domain models. The main requirements are (a) a well-defined syntax; (b) a well-defined semantics; (c) efficient reasoning support; (d) sufficient expressive power; and (e) convenience of expression. Web Ontology Language (OWL) has been designed to meet these needs. OWL is part of the growing stack of World Wide Web Consortium (W3C) recommendations related to the Semantic Web [14].

The example of ontology is presented with the recognition of human activities [15]. The ontological approach for activity modeling is composed of a knowledge engineering task to define the formal semantics of human activities by means of the operators of the ontological language. Each activity is defined as a specialization of the abstract *Activity* class. For instance, a *SocialActivity* can be defined as an activity having more than one actor. Activities are arranged in a hierarchical fashion. The part of the hierarchical ontology of social activities is shown in Fig. 1. Sub-activities are specifications of their parent activity. For instance, *TeaParty* can be defined as ''a specialization of *FriendlyMeeting* in which the actors are sipping tea in the afternoon''. Ontological reasoning is used to recognize that a user is performing a certain activity [16].

OWL is divided into three increasingly expressive sub-languages OWL-Lite, OWL-DL and OWL-Full. OWL-Lite was originally intended to support those users primarily needing a classification hierarchy and simple constraints. OWL-DL was designed to provide the maximum expressiveness possible while retaining computational completeness, decidability, and the availability of practical reasoning algorithms. OWL-DL includes all OWL language constructs, but they can be used only under certain restrictions. OWL-DL is so named due to its correspondence with description logic, a field of research that has studied the logics that form the formal foundation of OWL. OWL-Full is based on different semantics from OWL-Lite or OWL-DL and was designed to preserve some compatibility with RDF Schema. OWL-Full allows an ontology to augment the meaning of the predefined (RDF or OWL) vocabulary. There is no satisfactory implementation of OWL-Full in any reasoning software. OWL-DL is most often used because it provides maximum expressiveness [16].

Ontology can address the following three issues in a smart home.

- Dynamically compose service plans to fulfill the requested functions' base on available devices.
- Automatically supply and rank a list of functions that can be supported by a set of available devices.
- Dynamically adjust the system according to the environment and resident information.

In our smart home solution, ontology is introduced to deal with the three issues stated above based on the semantic information.

In this paper, we present an ADL ontology for smart homes because the ontology-based approach allows us to construct a knowledge base containing information on the certain environment that context. Use of ontology also allows us to create very flexible models on which deductions and reasoning can be done.

2.2. Smart homes

The recent emergence of ubiquitous environments, such as smart homes, has enabled the housekeeping, assistance and monitoring of chronically ill patients, persons with special needs or elderly in their own home environments in order to foster their autonomy in the daily living life by providing the required service when and where needed [2,17]. By using such technology, we can reduce costs considerably and alleviate healthcare systems. However, many issues related to this technology were raised such as activity recognition, person identification, assistance, and monitoring.

Activity recognition in smart environments is gaining increasing interest among researchers in ubiquitous computing and healthcare. Automatic recognition of activities is an important and challenging task. One of the typical applications in healthcare systems is the assistance and monitoring of the ADLs for the elderly and for people with special needs and to provide them with the appropriate services [3].

Several research works have been done, and several models have been proposed to recognize activities for smart environments. B. Chikhaoui et al. [3] proposed a new approach based on frequent pattern mining principle to extract frequent patterns in the datasets collected from different sensors disseminated in a smart environment. This approach adopted a hierarchical representation of activities and generated patterns for each activity model. In order to recognize activities, a mapping function is used between the frequent patterns and the activity models. They introduced only a new approach based on frequent pattern mining to recognize daily living activities.

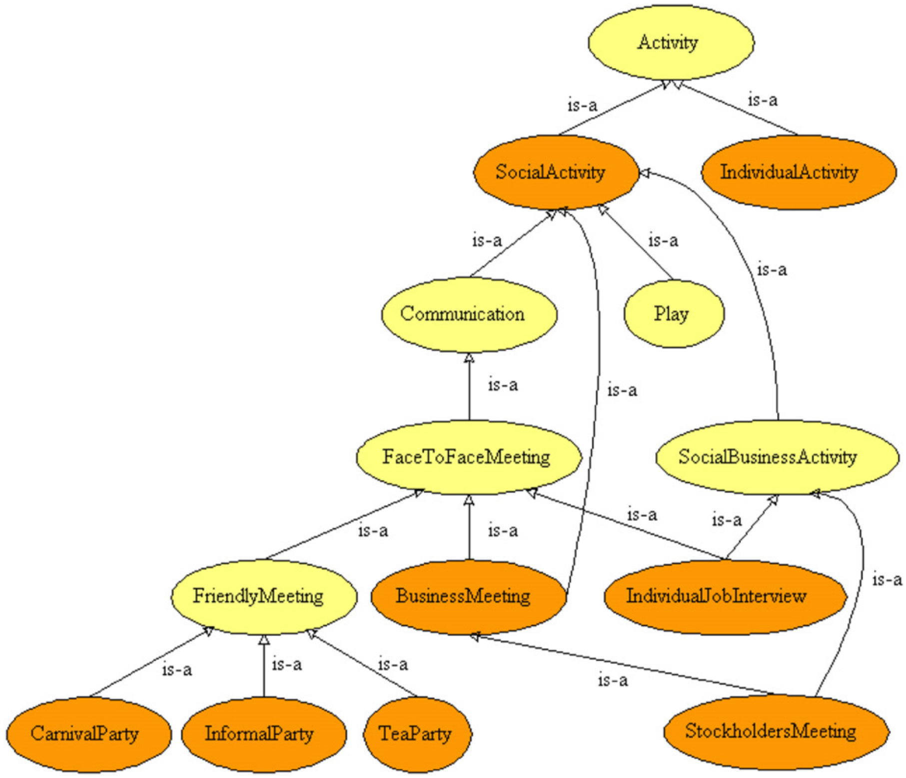

Fig. 1. Part of the ontology of social activities.

X. Hong and C.D. Nugent [4] presented HomeADL that addressed the issues associated with the heterogeneous nature of storage and distribution of the data within the smart environment community. They designed only the HomeADL concepts as series of hierarchical data trees: ADL, Activity and Action trees, but had not suggested for the implementation approach of HomeADL.

More recently, ontology-based modeling and representation has been applied in pervasive computing and in particular ambient assisted living. J. Xu [6] proposed an ontology-based framework to facilitate the automatic composition of appropriate applications. The system composed appropriate services depending upon the available equipments in each individual household automatically. The proposed approach was limited to the devices and their function ontology for home appliances.

P.A. Valiente and A. Lozano-Tello [7] presented IntelliDomo, an ontology-based AmI system for the control of domestic systems. It used domestic components and its state values represented as instances of an ontology and took advantage of the power of the production rules specified by the user in order to change the state of the system components in real time. IntelliDemo had the ability of making decisions and reacting to the changes that arise in the local home automation system elements, but it had not suggested the smart-phone based services through wireless Internet.

L. Chen and C. Nugent [8] introduced a systematic approach to providing situation-aware ADL assistances in a smart home environment. The approach makes use of semantic technologies for sensor modeling, fusion and management, thus creating machine understandable and process-able situational data. It exploits intelligent agents for interpreting and reasoning semantic situational data to enhance situation-aware decision support for cognitive assistance. But the proposed approach was implemented and evaluated only the kitchen ADL class hierarchy from a real world ADL assistance context. An earlier version of our paper [9] presented the ADL Recognition Method (ARoM). However, we expand the earlier version by the modification of ontology, the addition of rule reasoning for the service of AmIApplication in this paper. We also verify and evaluate the constructed ADL ontology for RADL.

More recently, ontology-based modeling and representation has been applied in pervasive computing and in particular ambient assisted living. The ontology-based activity recognition approach offers a feature that ontological ADL models can capture and encode rich domain knowledge and heuristic in a machine understandable and process-able way. Table 1 shows the functional comparison of the proposed RADL with other ontology-based ADL recognition methods for smart homes.

3. Ontology for RADL

The use of Ambient Intelligence (AmI) is one of the areas which are rapidly gaining importance in the application of intelligent systems in companies and homes. Ubiquitous computing suggests that computer and electric systems should be integrated into a physical environment and become part of it. AmI systems have sensors able to collect information in the environment.

These systems are usually knowledge-based systems containing the specification of domestic elements and they are based on production rules that represent the system's reasoning elements. A correct way of representing domestic systems and behavioral rules is through the use of ontology and the concepts established on the ontology. In this paper, RADL, the AmI system for smart homes, is based on the apartments for elderly or persons who live alone in South Korea. Each apartment provides a means of independent living. A typical layout of an apartment is shown in Fig. 2, where position sensors in each room have the ability to monitor the movement of a person through the home environment. The contact sensors detect if the window or the door has been opened or closed, and device sensors within electronic devices detect if the electronic devices have been turned on/off. Given the vast amount of information which may be generated from these sensors, it is necessary to discriminate between normal and abnormal situations.

Table 1
Functional comparison on ontology-based ADL recognition methods for smart homes.

Classification criteria		Methods			
		RADL	Xu [6]	IntelliDomo [7]	Chen [8]
Knowledge representation		OWL-DL	Ontology design	OWL-DL	SemWeb Semantic library
Ontology verification		OntoCheck, metric-based approach	None	None	None
Reasoning engine		Jess	None	Jess	Euler proof mechanism
	Monitor	Yes	Yes	None	None
	Comfort	Yes	Yes	Yes	None
Services	Housework	Yes	Yes	None	Yes
	Healthcare	None	None	None	None
	Message	Yes	None	None	None

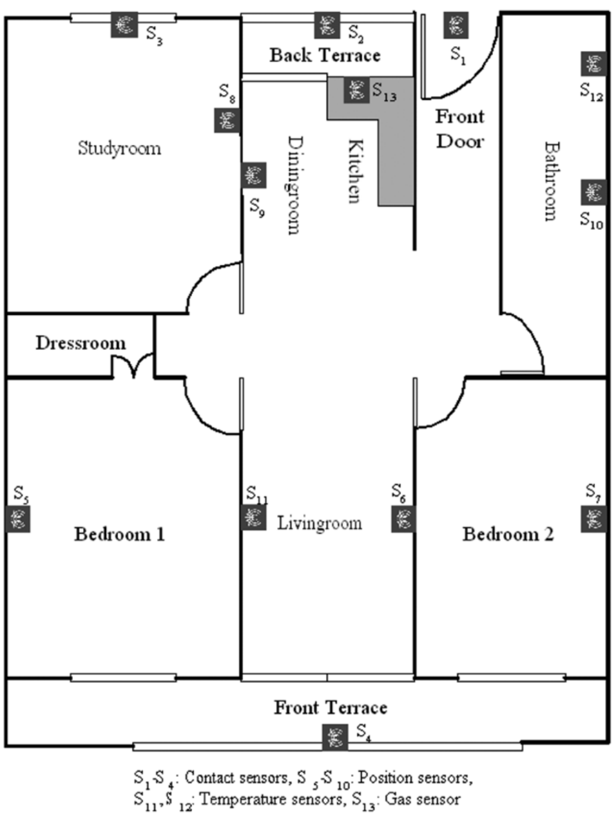

Fig. 2. Layout of space with various sensors to support independent living.

We design RADL, an ontology-based smart home system that discovers and monitors patterns of ADLs. Fig. 3 shows the overall architecture of an RADL system. A typical smart home environment contains sensors to detect the context and services to be invoked by users. Users interact with the system via user devices such as smart phones. The abstraction component maps the sensor data from various sensors to context information with a mapping function such as fuzzy membership functions. The context information is processed by the context provision component, which is a complex event processing system, producing formatted context events.

The service management stores service descriptions, monitors and invokes home services. Service registration and un-registration are also performed by this component. The services of a smart home environment are divided roughly into three categories: *DailyLifeService*, *SafetyService* and *MessageService*. The *DailyLifeService* has the actuator operations such as *AirConditionerOn*, *AirConditionerOff*, *CurtainOpenning*, *CurtainClosing*, *HeatingOn*, *HeatingOff*, *LightingOn*, *LightingOff*, *WindowOpening* and *WindowClosing*. *SafetyService* has *FireAlarm* and *TrespassingAlarm*. Also, *MessageService* is divided into two subclasses: *NoticeMessage* and *WarningMessage*. *NoticeMessage* has four types of messages: *SleepingMessage*, *WakeUpMessage*, *OutGoMessage* and *ComeHomeMessage*. Also, *WarningMessage* has two types of messages: *FireMessage* and *IntrusionMessage*.

The change detection decides when to trigger activity or service discovery. The ontology base component is introduced to support the semantic discovery for location, device and environment domain in smart homes. The reasoning component discovers the activity for a person and the appropriate service for a present situation. Sometimes the service management component forwards an ADL message to children or close relatives who live elsewhere through mobile networks. After the user identity certification process, the resident or the person who received the message can invoke a home service by sending a command to the service management component.

In order to infer the ADL of a smart home from temporal contexts, RADL is defined in this paper. In RADL, the semantic web ontology is used to represent the temporal contexts. The context model for the smart home is defined by OWL ontology, and the model is implemented by the Protégé tool, a graphical editor.

Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats [18].

The ontology model for RADL is shown in Fig. 4. It is composed of seven domains: *AmIApplication*, *Device*, *DeviStatus*, *Location*, *Person*, *Sensor* and *SensStatus* to represent the knowledge base in smart home systems. The *AmIApplication* ontology describes the concepts related to the ambient intelligent applications for the smart home. The *AmIApplication* class consists of two subclasses: *Activity* and *Service*. Also, the *Service* subclass consists of three subclasses: *DailyLifeService*, *SafetyService* and *MessageService*. The *Sensor* ontology describes the concepts related to various sensors which are installed in the smart home. The *Sensor* class consists of three subclasses: *ContextSensor*, *DeviceSensor* and *Timer*, where the *ContextSensor* subclass has four individuals that are instances of concepts: *Humidity*, *Light*, *Noise* and *Temperature*. Also, the *SensStatus* class consists of three subclasses: *ContextStatus*, *TimeDuration* and *TimeInstance*.

The OWL-based ontology for RADL has 98 instances, but Fig. 5 represents the instances of both the *Activity* subclass and the *DailyLifeService* subclass of the *AmIAppication* class in the ADL ontology.

Properties are binary relations on individuals, i.e. properties link individuals together. Table 2 shows the object properties in the OWL-based ontology for RADL. For example, the property *locatedIn* might link the domain class *Person* to the range class *Location*. Properties can be limited to having a single value i.e. to being *functional*. They can also be either *transitive* or *symmetric*. For example, the *inverse* of *hasDevice* is *hasSensor*.

Although on-the-shelf ontology editors' functionality is sufficient for daily ontology editing tasks, some clean-up checks on

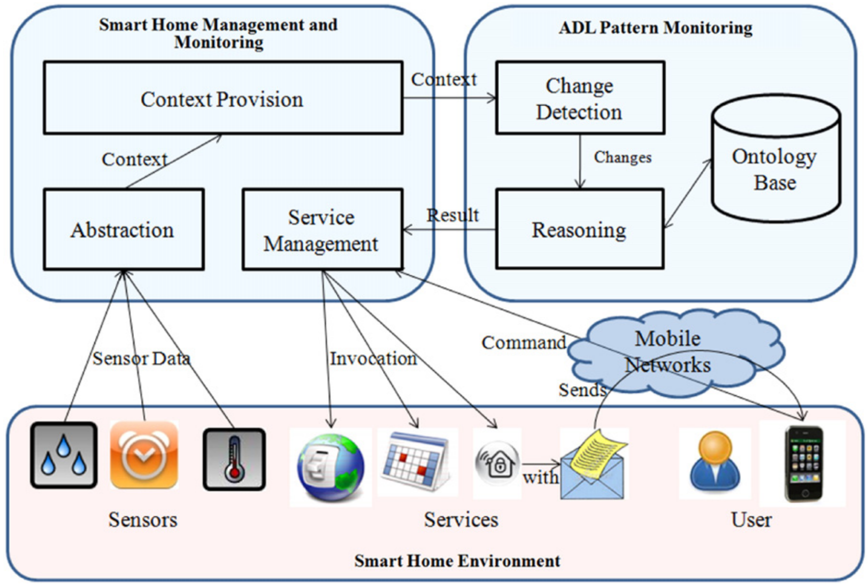

Fig. 3. System architecture of RADL.

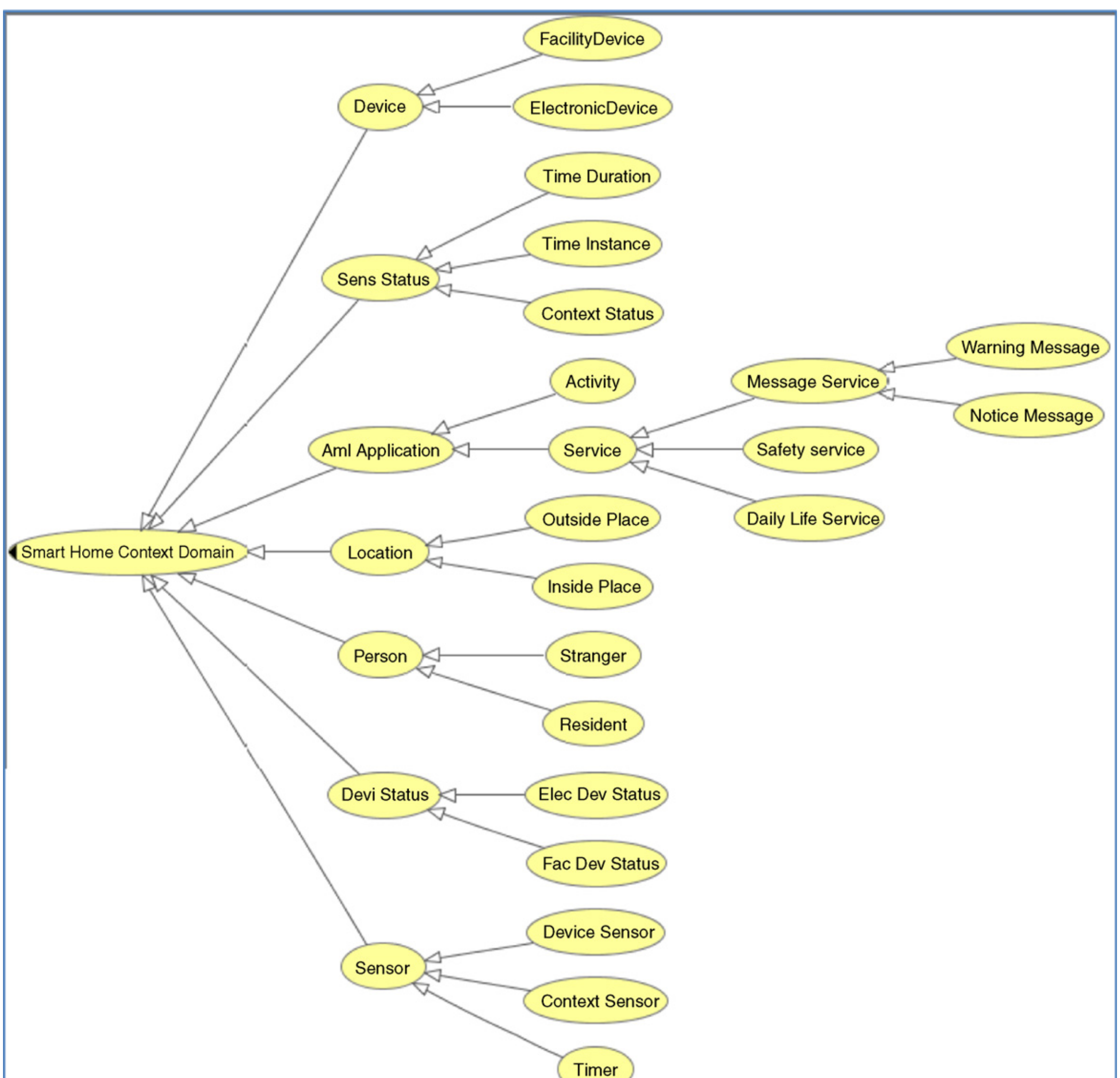

Fig. 4. OWL-based ontology for RADL.

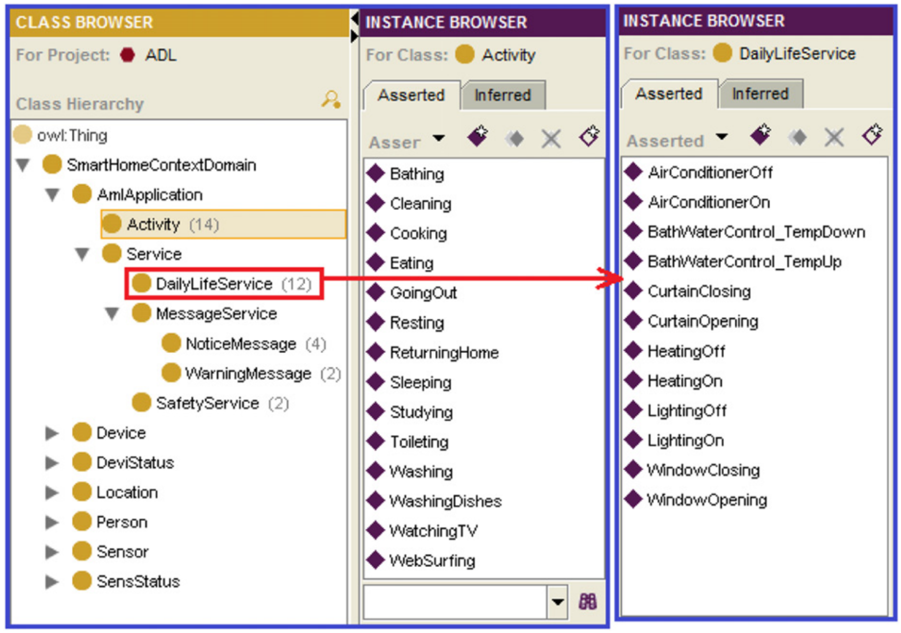

Fig. 5. Instances of *Activity* and *DailyLifeService* subclasses.

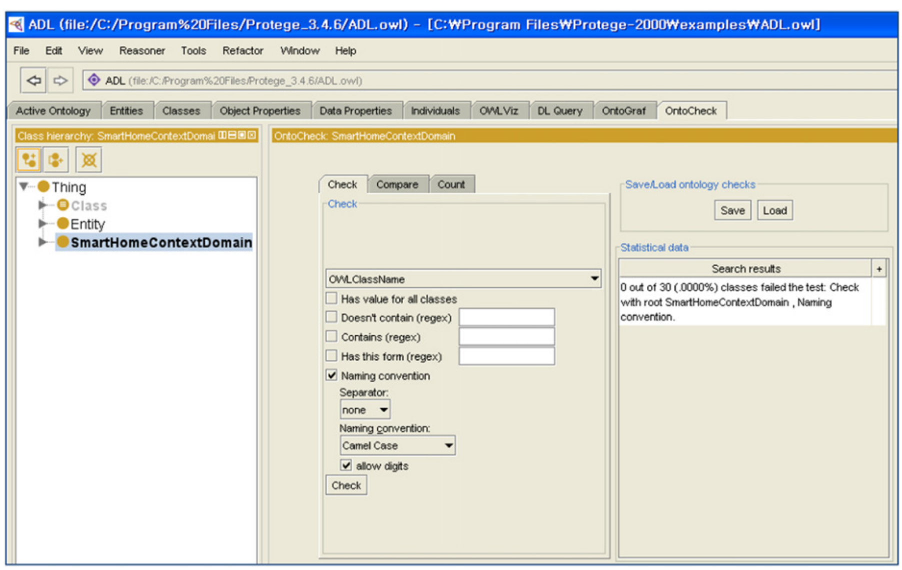

Fig. 6. The Check pane within the OntoCheck Tab for the ontology of RADL.

Table 2
Object properties in the OWL-based ontology for RADL.

Object property	Domain	Range	Property characteristics
calls	Person	Service	Functional
has	Person	Activity	Functional
hasDevice	Sensor	Device	InverseFunctional
hasDuration	Timer	TimeDuration	Functional
hasSensing	Sensor	ContextStatus	Functional
hasSensor	Device	Sensor	InverseFunctional
hasStatus	Device	DeviStatus	Functional
hasTime	Timer	TimeInstance	Functional
locatedIn	Person	Location	Functional
uses	Person	Device	Functional
with	Person	AmIApplication	MessageService	Functional

the ontology could complement Protégé in a useful way. We use the OntoCheck plugin [19,20] that checks certain properties of an active OWL ontology and allows for amendments in the areas of metadata analysis and naming conventions. The OntoCheck plugin provides a new tab that is organized in the three panels: Check, Compare and Count. Fig. 6 shows the Check pane within the OntoCheck Tab that displays the specification of a test for CamelCase—no separator naming convention for the active ADL ontology.

OntoCheck is carried out on the ontology for RADL. Absolute counts and the percentages of found classes violating the checks were measured. The result is shown in Table 3, where that 'Compare in different classes (OWL ClassName Is contained)' is 0 indicates 0 out of 30 (0.0%) classes failed the test: compare with root SmartHomeContextDomain in different classes OWL ClassName, is contained. The verification of the ontology for RADL is confirmed by Table 3.

With this growth in the number of ontologies, there have been some attempts to study the different approaches and tools for ontology evaluation and validation. Metric-based approaches to evaluate ontologies offer a quantitative perspective of ontology quality. Metric-based approaches divide the metrics into two

Table 3
OntoCheck test cases and detected quantified violations.

Target node	Check or compare	Violations (%) or counts (number)
SmartHomeContextDomain	Check Naming convention (Camel Case)	0.0%
	Compare in different classes (OWL ClassName Equals)	0
	Compare in different classes (OWL ClassName Is contained)	0
	Compare property 'a', 'b' of the same class (OWL ClassName Contains owl:deprecated)	0
	Compare property 'a', 'b' of the same class (OWL ClassName Contains owl:incompatible with)	0

Table 4
The summary results of schema and instance metrics for RADL.

Category	Metrics	Equations	Values
Schema	RR (Relationship Richness)	$RR = \|P\|/(\|H\| + \|P\|)$	$RR = 11/(29 + 11) = 0.275$
Instance	CR (Class Richness)	$CR = \|C'\|/\|C\|$	$CR = 18/30 = 0.6$

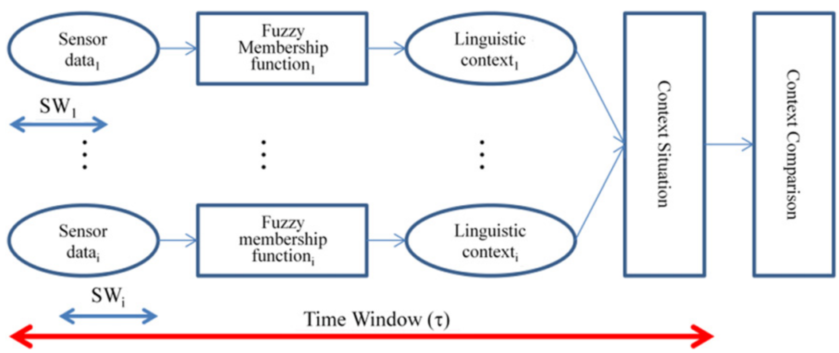

Fig. 7. Sensor data stream processing of RADL.

related categories: the schema metrics and instance metrics. The schema metrics address the design of the ontology. Although we cannot definitely know if the ontology design correctly models the domain knowledge, metrics in this category indices the richness, width, depth, and inheritance of an ontology schema design. Instance metrics evaluate the placement of instance data within the ontology and the effective usage of the ontology to represent the knowledge modeled in the ontology [21].

Table 4 shows the summary results of schema and instance metrics for RADL, where H, P, C and C' represent the number of inheritance relationships, the number of non-inheritance relationships, the number of classes and the number of non-empty classes, respectively.

From the results of Table 4, we know that many of the relationships are inheritance relationships and the data in the knowledge base represents most of the knowledge in the schema.

4. ADL reasoning

A semantic reasoner, reasoning engine, rule engine, or simply a reasoner is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes an inference engine, by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language, and often a description language.

A large number of rule engines work well with Java, and many are available as open source software. Some of the most popular engines include Jess, Algernon [22] and SweetRules [23]. We choose Jess as the semantic reasoner for ADL reasoning.

The Jess (Java Expert System Shell) system [24,25] consists of a rule base and an execution engine. A Jess engine running inside the Protégé framework is the basis for the JessTab integration model. Because Protégé and Jess are implemented in Java, we can run them together in a single Java virtual machine. This approach lets us use

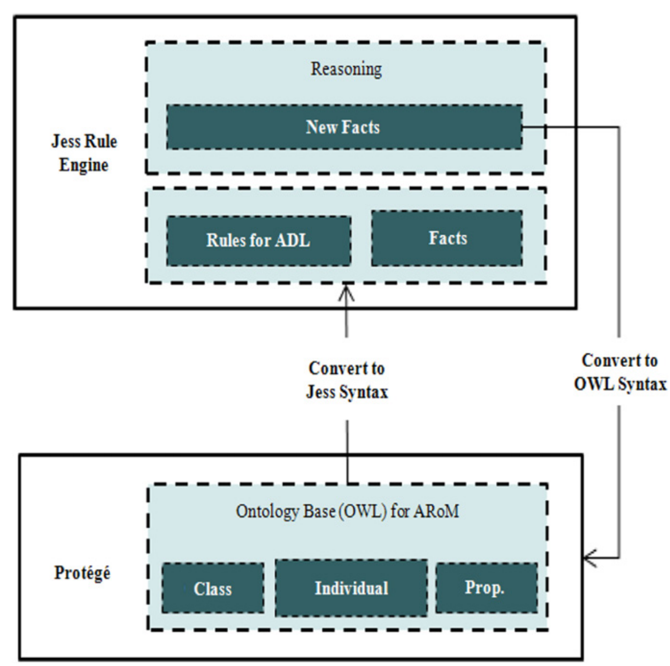

Fig. 8. OWL/Jess integration structure for ADL reasoning.

Jess as an interactive tool for manipulating. JessTab integrates the knowledge representation models by mapping Protégé instances to Jess facts. This representation of instances as facts in Jess lets us effectively write Jess rules that match instance patterns. In the Protégé frame model, classes, slots, and facets are themselves instances. So, it is sufficient to map Protégé instances to facts. This mapping approach fits well with both the Protégé and Jess models.

Sensors that are embedded in the smart home have a varying sampling rate which depends on the human activity [26]. Suppose

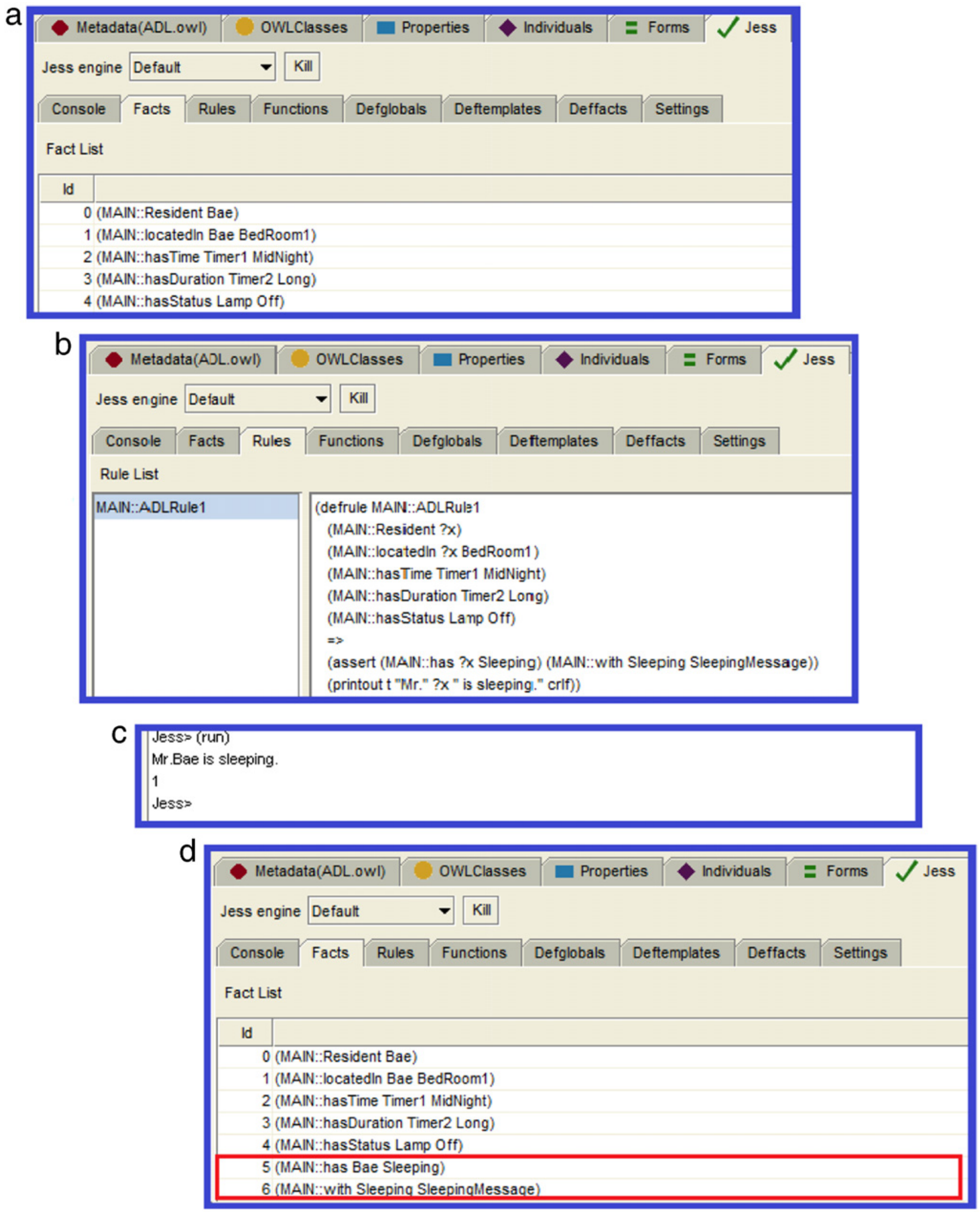

Fig. 9. Example of the rule reasoning for *Activity* of *AmIApplication*.

that a sequence of state changes along the timeline, a situation at a specific time point can be described as the accumulation of states which occurred before that particular time [8]. Accordingly, we categorize context data into three layers according to the degree of abstraction and semantics: the sensor layer, the context layer and the context situation layer, as shown in Fig. 7. The sensor layer is the source of context data. Sensor data is the output set of the sensor layer at a sensor window (SW). The context layer serves as an abstraction between the physical world and semantic world. Context data is the linguistic variable that corresponds to the numerical data from a sensor by a membership function. The membership function uses a fuzzy set. The context situation layer is conceptualized as a snapshot of states at a specific time window in a physical or conceptual environment. A situation at a specific time window τ can be described as the accumulation of states which occurred within that particular time window.

For example, assuming that resident position, gas stove sensors and time duration in our smart apartment are three states as shown in Fig. 2. We assume ‘ON’ for the position sensor S_9 of dining room, ‘ON’ for the gas stove sensor and ‘15 MIN’ for time duration. The numerical data ‘15 MIN’ is mapped to the linguistic context data ‘M (medium)’ by a fuzzy membership function for the time duration. From these context data, it is inferred that the current context situation is ‘Cooking’.

Then, the change detection component of RADL compares the context situation of the current time window with the context situation of the previous time window. If the current context situation differs from the previous context situation, the reasoning component of RADL is invoked. Next, the service management component invokes a service or sends a message according to the result of the reasoning.

Fig. 8 shows the OWL/Jess integration structure for ADL reasoning. Once the RADL OWL concepts have been represented in Jess, the Jess execution engine can perform inference. As ADL rules fire, new Jess facts are inserted into the fact base. Those facts can be used in further inference. When the inference process completes, these facts must be transformed into OWL knowledge.

Fig. 9 shows an example of rule reasoning for *Activity* of *AmIApplication*. In this example, the activity of a resident is inferred when the resident is sleeping in the bedroom at midnight, the Jess

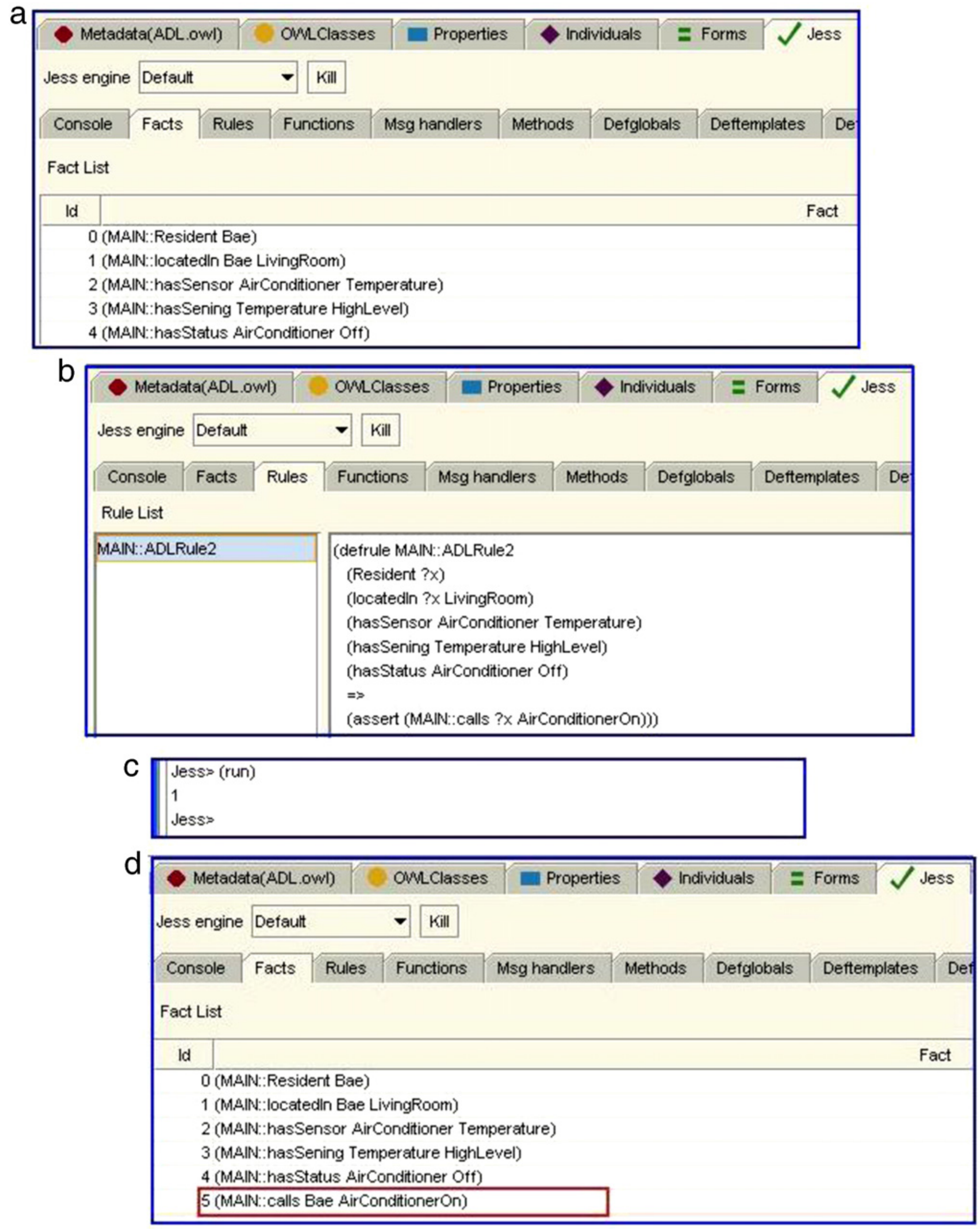

Fig. 10. Example of the rule reasoning for *Service* of *AmIApplication*.

system has currently five facts that are represented in Fig. 9(a), the Jess engine performs the ADL rule that is represented in Fig. 9(b), and the Jess system displays the inference result message for the ADL rule, Fig. 9(c). After the ADL rule has been performed, a new fact is generated by the Jess engine, the new fact 'has Bae Sleeping' and 'with Sleeping SleepingMessage', Fig. 9(d), are inserted into the fact base, and the short message that is generated by the inference is sent to children or close relatives who live elsewhere through wireless networks.

Fig. 10 shows an example of rule reasoning for *Service* of *AmIApplication*. In this example, when a resident enters a smart home with unpleasant high temperature, the smart services switch on the air conditioner automatically. The Jess system has currently five facts that are represented in Fig. 10(a), the Jess engine performs the ADL rule, Fig. 10(b), and the Jess system displays the inference result message for the ADL rule, Fig. 10(c). After the ADL rule has been performed, a new fact is generated by the Jess engine, the new fact 'calls Bae AirConditionerOn' (the part of rectangular box in Fig. 10(d)) is inserted into the fact base. Then the actuator in the smart home runs the air conditioner in the living room.

5. Conclusions and future work

In this paper, we have suggested RADL that discovers and monitors the patterns of ADLs in sensor equipped smart homes. The RADL consists of two components: smart home management monitoring and ADL pattern monitoring. As a result of this research, the ontology model for RADL has been designed and the prototype system of RADL has been implemented by using Protégé and Jess tools. We have not only verified the ontology for RADL through OntoCheck but have also performed the evaluation of the ontology for RADL through a metric-based approach. We found that many of the relationships were inheritance relationships and the data in the knowledge base represented most of the knowledge in the schema.

In future, we will work on a fault tolerant ADL recognition method, which would combine our methodology as proposed in Section 4 with the generalization capabilities of approaches such as, e.g., rough-fuzzy attribute reduction and rule simplification techniques discussed in [27]. We will also work on the full version of the proposed RADL framework in order to support all daily services relevant to the smart home infrastructure.

References

[1] F. Sadri, Ambient intelligence: a survey, ACM Computing Surveys 43 (4) (2010) 1–66.
[2] P. Rashidi, D.J. Cook, L.B. Holder, M. Schmitter-Edgecombe, Discovering activities to recognize and track in a smart environment, The IEEE Transactions on Knowledge and Data Engineering 23 (4) (2011) 527–539.
[3] B. Chikhaoui, S. Wang, H. Pigot, A frequent pattern mining approach for ADLs recognition in smart environments, in: International Conference on Advanced Information Networking and Application, Biopolis, Singapore, March 22–25, 2011.
[4] X. Hong, C.D. Nugent, HomeADL for adaptive ADL monitoring within smart homes, in: Annual International IEEE Engineering in Medicine and Biology Society Conference, Vancouver, Canada, August 21–24, 2008.
[5] M.R. Alam, A review of smart homes—past, present, and future, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 42 (6) (2012) 1190–1203.
[6] J. Xu, Ontology-based smart home solution and service composition, in: Int. Conf. of Embedded Software and Systems, HangZhou, Zhejiang, China, May 25–27, 2009.
[7] P.A. Valiente-Rocha, A. Lozano-Tello, Ontology-based expert system for home automation controlling, in: 23th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, Cordoba, Spain, June 1–4, 2010.
[8] L. Chen, C. Nugent, Situation aware cognitive assistance in smart homes, Journal of Mobile Multimedia 6 (3) (2010) 263–280.
[9] I.-H. Bae, H.G. Kim, An ontology-based ADL recognition method for smart homes, in: International Conference on Future Generation Communication and Networking, JeJu, Korea, December 8–10, 2011.
[10] T. Mori, A. Fujii, M. Shimosaka, H. Noguchi, T. Sano, Typical behavior patterns extraction and anomaly detection algorithm based on accumulated home sensor data, in: Proceedings of the Future Generation Communication and Networking, Jeju-Island, Korea, December 6–8, 2007.
[11] R. Studer, R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, Data & Knowledge Engineering 25 (1–2) (1998) 161–197.
[12] M. Viinikkala, Ontology in information systems, 2004. http://www.cs.tut.fi/~kk/webstuff/Ontology.pdf.
[13] F. Ay, Context modeling and reasoning using ontologies, 2007. http://www.ponnuki.de/cmaruo/cmaruo.pdf.
[14] G. Antoniou, F.v. Harmelen, Web ontology language: OWL, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, 2003, pp. 67–92.
[15] D. Riboni, C. Bettini, OWL 2 modeling and reasoning with complex human activities, Pervasive and Mobile Computing 7 (3) (2011) 379–395.
[16] W3C, OWL web ontology language guide, 2004. http://www.w3.org/TR/owl-guide/.
[17] W. Qin, Y. Shi, Y. Suo, Ontology-based context-aware middleware for smart space, Tsinghua Science and Technology 2 (6) (2007) 707–713.
[18] H. Eriksson, Using TessTab to integrate Protégé and Jess, IEEE Intelligent Systems 18 (2) (2003) 43–50.
[19] D. Schober, I. Tudose, V. Svatek, M. Boeker, OntoCheck: verifying ontology naming conventions in Protégé 4, in: OBML 2011 Workshop Proceedings, Berlin, Germany, October 6–7, 2011.
[20] D. Schober, I. Tudose, V. Svatek, M. Boeker, Checking class labels against naming conventions: first experience with the OntoCheck Protégé plugin, in: 3rd International Conference on Biomedical Ontology, Graz, Austria, July 21–25, 2012.
[21] S. Tartier, I.B. Arpinar, A.P. Sheth, Ontological evaluation and validation, Theory and Applications of Ontology: Computer Applications 2 (2010) 115–130.
[22] Algernon—rule-based programming. http://algernon-j.sourceforge.net/.
[23] SweetRules: tools for semantic web rules and ontologies, including translation, inferencing, analysis, and authoring. http://sweetrules.projects.semwebcentral.org/.
[24] M. Horridge, A practical guide to building OWL ontologies using Protégé 4 and CO-ODE tools. http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_1.pdf.
[25] J. Mei, E.P. Bontas, Reasoning paradigms for OWL ontologies, Technical Report B-04-12, Department of Information Science, Peking University, 2004.
[26] N.C. Krishnan, D.J. Cook, Activity recognition on streaming sensor data, Pervasive and Mobile Computing 8 (5) (2012) 730–750.
[27] C. Cornells, R. Jensen, G. Hurtado, D. Ślęzak, Attribute selection with fuzzy decision reducts, Information Science 180 (2) (2010) 209–224.

Ihn-Han Bae received his Ph.D. degree in Computer Engineering from Chung-Ang University, Korea, in 1990. Now he is a professor of school of information technology at Catholic University of Daegu. He worked as a post-doctoral at the Ohio State University between 1996 and 1997. He worked as a visiting scholar at the Old Domination University between 2002 and 2003 and between 2009 and 2010. His research interests include intelligent smart application, vehicular networks, cloud computing and mobile multimedia.